Predicción de Equipo Ganador en el Baloncesto

Universidad Carlos III de Madrid

Autor: Jorge Morate Vázquez

Tutor: Antonio Berlanga de Jesús

Doble Grado en Ingeniería Informática y Administración y Dirección de Empresas

Abstract

El objetivo principal de este proyecto es poder predecir y clasificar correctamente qué equipo de la NBA (National Basketball Association) ganará un partido concreto, en base a datos previos y objetivos de la temporada regular 2015/2016.

Paralelamente, se realizará otro experimento para predecir qué equipo será el primero en anotar una canasta en cada partido. El objetivo de esta observación es el de comprobar si realmente este hecho ocurre de manera no predecible.

Para llevar a cabo ambas experimentaciones, se van a emplear distintas técnicas de aprendizaje automático con el fin de poder evaluar los resultados desde diferentes enfoques.

Con el mejor modelo clasificador, se explotará el sistema sobre un conjunto de partidos apostando dinero en función de las predicciones del mismo.

Palabras clave: Aprendizaje Automático, Weka, Predicción de resultados, NBA, Atributos, Mineria de datos, Apuestas deportivas.

Agradecimientos

A mi tutor Antonio Berlanga, por su guía, consejo y por ayudarme a trazar el camino de este proyecto desde un punto de vista experto.

A mis amigos, y especialmente a mis compañeros de la Hack Crew. Alberto, Alejandro, Álvaro y Eduardo por 5 años magníficos.

A Víctor, por ayudarme a entender el campo del Aprendizaje Automático con paciencia. Pero sobretodo, gracias por haber compartido conmigo cinco años de proyectos y largas noches de trabajo desde el humor, y las ganas de ser mejores.

A Diego, por compartir conmigo la ilusión de este proyecto y aportar nuevas ideas al mismo. Todo el mundo necesita un amigo como tú.

A mi familia, especialmente a mi madre y a mi padre por permitirme llegar hasta este punto.

A María, por todo.

Jorge Morate Vázquez

Junio 2016

Contenidos

Índice de Tablas

Índice de Ilustraciones

Índice de gráficas

Acrónimos

NBA – National Basketball Association

RNA – Red de Neuronas Artificiales

SPN – Neural Perceptron Simulator

SVM – Support Vector Machine

JRip – Java Repeated Incremental Prunning

CRISP-DM - Cross Industry Standard Process for Data Mining

1. Introducción

1.1 Objetivos, motivación y contexto.

En la historia del deporte, y especialmente a partir del siglo XX, los resultados y estadísticas han sido recogidos con cuidado con diferentes fines. Gracias a esta recogida, se tiene constancia objetiva de los rendimientos de los equipos y jugadores a lo largo del tiempo. Además se pueden establecer records e hitos de distinta índole. Especialmente, en el baloncesto, la recogida de datos es importante y ampliamente utilizada. El rendimiento de los jugadores en este deporte es fácil de obtener numéricamente, lo que permite comparar el desempeño de distintos equipos desde los distintos puntos de vista que ofrece una cantidad tan amplia de atributos medibles.

Por otro lado, la informática, entre sus distintas y numerosas aplicaciones, ha desarrollado maneras de poder predecir acontecimientos en base a una recogida previa de datos. Estas predicciones se utilizan en numerosos campos como la medicina, la meteorología o la biología, y en algunas de ellas los resultados son realmente positivos. Para ello se emplean técnicas de aprendizaje automático, clasificadas en supervisadas y no supervisadas.

La fusión entre el baloncesto, y la informática es notable, y esta última dota de grandes ventajas a la primera. Entre todas ellas, nos incube el buen soporte que ofrece la informática para recoger y consultar fácilmente las estadísticas históricas de este deporte, pero, principalmente, nos motiva la potencial capacidad que posee para predecir qué equipo ganará un partido en concreto. Es decir, nos interesa extrapolar al baloncesto los procedimientos de predicción llevados a cabo en otros campos.

Esta predicción puede resultar útil para un apostante de baloncesto, con el fin de obtener beneficio económico apostando en un partido concreto al equipo ganador basándose en este estudio. Sin embargo, este proyecto no se centrará únicamente en predecir quién ganará. En el campo de las apuestas de

baloncesto, existe la posibilidad de apostar sobre qué equipo será el primero en anotar una canasta. Las casas de apuestas ofrecen el mismo beneficio para los dos equipos que juegan un partido, por lo que se presupone que, a priori, no existen evidencias de que uno de los dos equipos tenga mayor probabilidad de ser el primero en anotar. Por ello, en este trabajo se va a realizar un estudio sobre la temporada regular 2015/2016 para tratar de encontrar atributos que puedan ayudar a predecir qué equipo será el primero en anotar en un partido concreto.

2. Estado del arte

En primer lugar se va a estudiar en qué punto histórico se encuentra la NBA actualmente, posteriormente la situación en la que se encuentra la tecnología en el campo de la predicción, y finalmente, los hallazgos obtenidos hasta la fecha producto de la fusión entre baloncesto e informática.

2.1 Baloncesto – NBA: formato y estadísticas.

El baloncesto es un deporte de equipo en el que dos equipos, compuestos por 5 jugadores cada uno, compiten por anotar una mayor cantidad de puntos que su oponente. Estos puntos se consiguen introduciendo un balón de 23 centímetros de diámetro en un aro situado a 3,05 metros del suelo y 45,7 centímetros de diámetro. En este deporte no existe el empate, ya que si al finalizar el periodo reglamentario las puntuaciones de ambos equipos son iguales, se jugarán prórrogas hasta que uno de los dos equipos resulte vencedor. Fue inventado en diciembre de 1891 por James Naismith, profesor canadiense en educación física en la YMCA, Springfield, Massachusetts, Estados Unidos. [1]

La Nationall Basketball Association, en adelante NBA, es la principal liga estadounidense y mundial de baloncesto profesional. Fundada en 1946, cuenta con 30 equipos con sede en los distintos estados de Estados Unidos, y en ciertas ciudades de Canadá. Estos 30 equipos se subdividen en aquellos con sede en la conferencia oeste y aquello situados en la conferencia este.

La competición se divide en dos etapas. Una primera conocida como temporada regular, en la que cada uno de los 30 equipos juega 82 partidos divididos a partes iguales entre local y visitante. Normalmente, esta fase comienza en el mes de octubre, y concluye en el mes de abril del año siguiente. Al finalizar esta fase, los 8 equipos de cada conferencia con mayor porcentaje de

victorias serán los que pasen a la siguiente etapa, quedando finalizada la temporada para los equipos restantes.

La segunda fase es conocida como los Playoffs, formada por cuatro rondas: Primera Ronda, Semifinales de Conferencia, Finales de Conferencia y Finales de la NBA, las cuales son jugadas por los campeones de cada Final de Conferencia.

Son numerosas las razones por las que esta liga resulta tan atractiva. En primer lugar, su sistema de límite salarial, y de elección de nuevos jugadores, la convierte en una de las ligas más igualadas del mundo. Los equipos que han quedado peor clasificados al final de una campaña, tendrán preferencia para elegir nuevos jugadores que quieren entrar en la liga en la próxima temporada. Además, los equipos no pueden superar cierto límite salarial. Esto implica que un equipo debe administrar cuidadosamente los salarios de sus jugadores, especialmente los de sus 2 o 3 jugadores con mayor rendimiento. Evidentemente, estos jugadores exigen un mayor salario y por tanto forman parte de un alto porcentaje del límite salarial de un equipo. Es por ello que estos jugadores son denominados los "jugadores franquicia", ya que en ellos está depositada la mayor parte de la confianza en el buen rendimiento del equipo. Precisamente, el cálculo de cuánto disminuye el rendimiento de un equipo si su jugador franquicia no juega un partido, resulta un tema atractivo para la informática.

Por otra parte, las estadísticas de la NBA son recogidas oficialmente por la propia NBA, y son consultables en su página web dedicada: stats.nba.com. Sin embargo, existen numerosas asociaciones, o colectivos que se dedican a post-procesar la base de datos oficial con diferentes fines. Algunos de estos fines son: Generar pequeños predictores de apuestas, consultar los datos en mejores interfaces de usuario, crear consultas a la base de datos de carácter curioso o hacer más interesantes las previas de los partidos mediante el análisis de los rendimientos de cada equipo.

2.2 La NBA en las apuestas deportivas

Las apuestas deportivas son la actividad de predecir resultados deportivos, invirtiendo dinero en un posible resultado con el fin de obtener un beneficio si la predicción es correcta. En el baloncesto, y concretamente en la NBA, los sistemas de apuestas son muy conocidos, ya que los apostantes creen tener mucha información debido a la cantidad de estadísticas extraíbles que ofrece este deporte.

A continuación, se muestra la interfaz para apostar en un partido de NBA a través de la casa de apuestas William Hill:

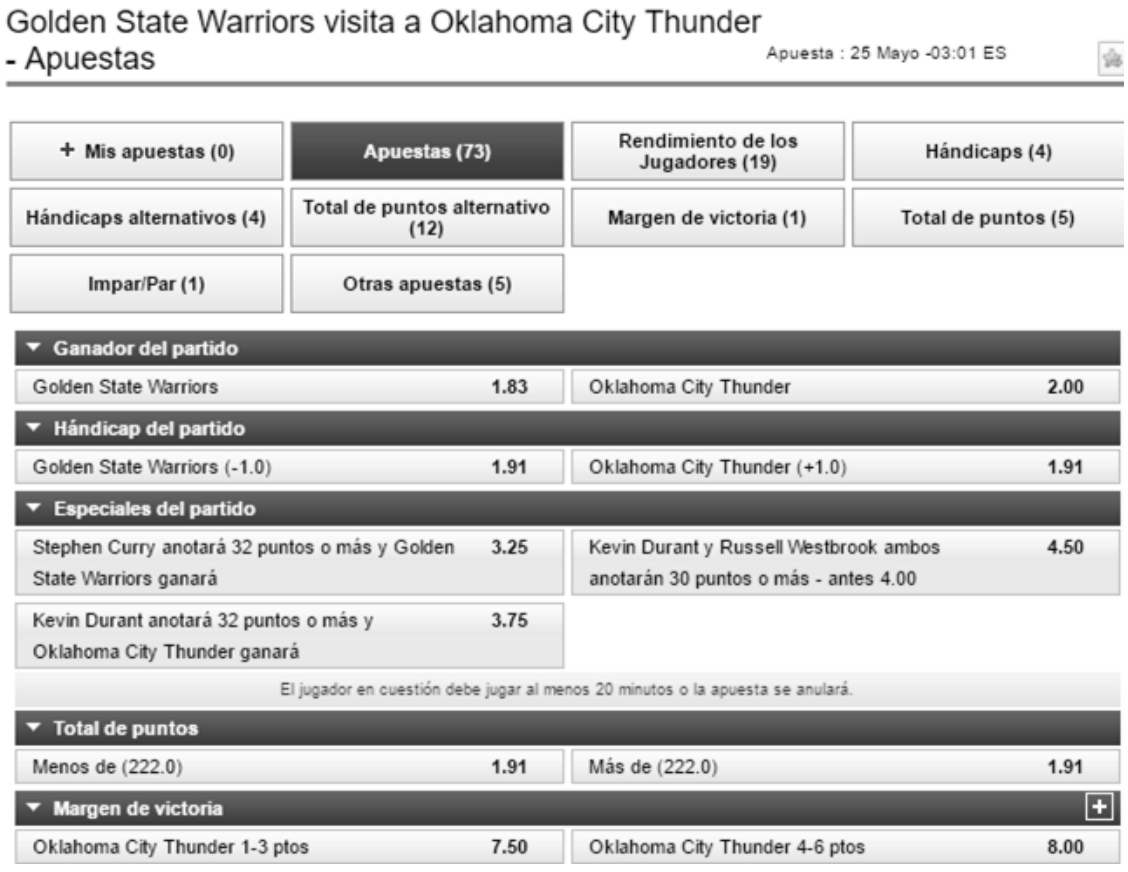


Ilustración 1: Mercado de apuestas para Golden State - Oklahoma City

Para conocer el funcionamiento de una apuesta simple, se fijará la atención en la apuesta "Ganador del partido". En ella se puede observar como

Golden State Warriors tiene una cuota de 1.83 y Oklahoma City Thunder tiene una cuota de 2.00. Las posibles ganancias si se apuestan 10 € son:

	Cuota Oklahoma: 2.00	**Cuota Golden State: 1.83**	**Beneficio Neto**
Si se apuesta a Oklahoma y Gana Oklahoma	2.00 x 10 = 20.00	-	20 – 10 = 10€
Si se apuesta a Golden State y Gana Golden State	-	1.83 x 10 = 18.3	18.3 – 10 = 8.3 €
Cualquier otro caso			-10 €

Tabla 1: Escenarios para apuesta simple (Golden State - Oklahoma City)

Las casas de apuestas fijan las cuotas en base a:

- El cálculo estadístico: Calculan la probabilidad de cada evento contando con distintos atributos como los rendimientos de los equipos o jugadores importantes lesionados entre muchos otros.
- El margen de beneficio que obtienen.

Con estos requisitos, la cuota final se establece mediante el método de cantidades implícitas [2]:

Tomando como ejemplo el partido entre Oklahoma City y Golden State, primero se calculan las probabilidades simples: $P = \frac{1}{\text{cuota}} \text{ x } 100$

$$P_{Oklahoma} = \frac{1}{2{,}00} \text{x } 100 = 50\%$$

$$P_{Golden\ State} = \frac{1}{1{,}83} \text{x } 100 = 54.54\%$$

Como se puede observar la suma de probabilidades es superior al 100%. Este superávit representa el margen de beneficio de la casa de apuestas.

$$Beneficio\ Casa\ Apuestas = \frac{1}{2{,}00} + \frac{1}{1{,}83} = 1.0464$$

$$Porcentaje\ de\ pagos = \frac{1}{1{,}0464} \text{x}\ 100 = 95.56\%$$

Es decir, por cada 100€ apostados en el partido, la casa de apuestas espera devolver el 95,56% a los apostantes.

Con estos datos ahora es posible calcular la probabilidad implícita (probabilidad simple multiplicada por porcentaje de pagos.):

$$P_{Oklahoma} = 0{,}5 \times 0{,}9556 = 47{,}78\%$$

$$P_{Golden\ State} = 0{,}5454 \times 0{,}9556 = 52.22\%$$

Se puede comprobar como ahora, las probabilidades sí suman 100%, y por tanto esta es la probabilidad real que otorgan a cada posibilidad para que ocurra.

Por otro lado, existen ciertas apuestas que las casas de apuestas consideran como equiprobables. En concreto, a continuación se muestra la apuesta "Equipo que marcará la primera canasta".

Ilustración 2: Apuesta - Equipo que marcará la 1ª canasta

Preguntarse si realmente estos hechos son equiprobables resulta interesante. Los estudios de eficiencia del mercado de las apuestas deportivas son similares a los de los mercados financieros. Ambos poseen muchos participantes, grandes cantidades de dinero, participantes informados y desinformados, fricciones de mercado, informaciones asimétricas, y, como muestra la evidencia, un fuerte factor psicológico. En las literaturas de los mercados financieros, se explica que

las ineficiencias del mercado deben ser validadas con ciertos test para confirmar si no son simples coincidencias periódicas formadas por la distribución de los datos. El mundo de las apuestas deportivas nos ofrece un escenario único para probar su eficiencia, ya que los retornos por apostar son conocidos con certeza antes de conocer el resultado final.[3] [4]

2.3 Minería de datos

La minería de datos es un proceso mediante el cual se halla información procesable y útil que se encuentra en algún otro conjunto de datos. Gracias a este proceso es posible descubrir patrones en grandes almacenamientos de información, yendo más allá del simple análisis.

A nivel tecnológico, la minería de datos ha resuelto dos grandes retos: extraer información interesante a partir de grandes conjuntos de datos, y utilizar técnicas para explorar, analizar, comprender e identificar patrones que ayuden a entender el entorno y nos ayude a tomar decisiones [5]. Su creciente uso se debe a diversas circunstancias:

- El aumento de recogida de datos sumado a la evolución de la capacidad de cómputo.
- El almacenamiento de los datos en data-warehouses, con el fin de tener acceso a bases de datos actualizadas y fiables.
- El aumento de la facilidad de acceso a la información.
- Sus aplicaciones a la economía globalizada, y el desarrollo de herramientas para llevar a cabo la minería de datos con interfaces cada vez más sencillas.

Paralelamente, la minería de datos ha unificado numeroso campos, tanto en su aplicación, como durante su proceso. El proceso es aplicable, entre otras

disciplinas, a las finanzas, al análisis de mercados, a la medicina, a las telecomunicaciones, a la seguridad, al análisis ambiental o a la química. Pero además, durante el proceso se relacionan campos como las bases de datos, la estadística, las redes neuronales artificiales, el aprendizaje automático, la computación paralela, la inteligencia artificial o el reconocimiento de patrones entre otros. [5]

Sin embargo, el concepto de minería de datos suele ser mal utilizado para referirse a cualquier manera de procesar datos a gran escala. La diferencia más notable entre la minería de datos y las estadísticas tradicionales reside en que la primera utiliza el análisis matemático para deducir los patrones que existen en los datos. Lo normal, es que estos patrones no se puedan deducir mediante exploraciones estadísticas tradicionales ya que las relaciones entre los datos son complicadas o porque la cantidad de los mismos es muy elevada. [6]

El proceso que sigue la minería de datos se detalla a continuación, y se apoya de la siguiente ilustración:

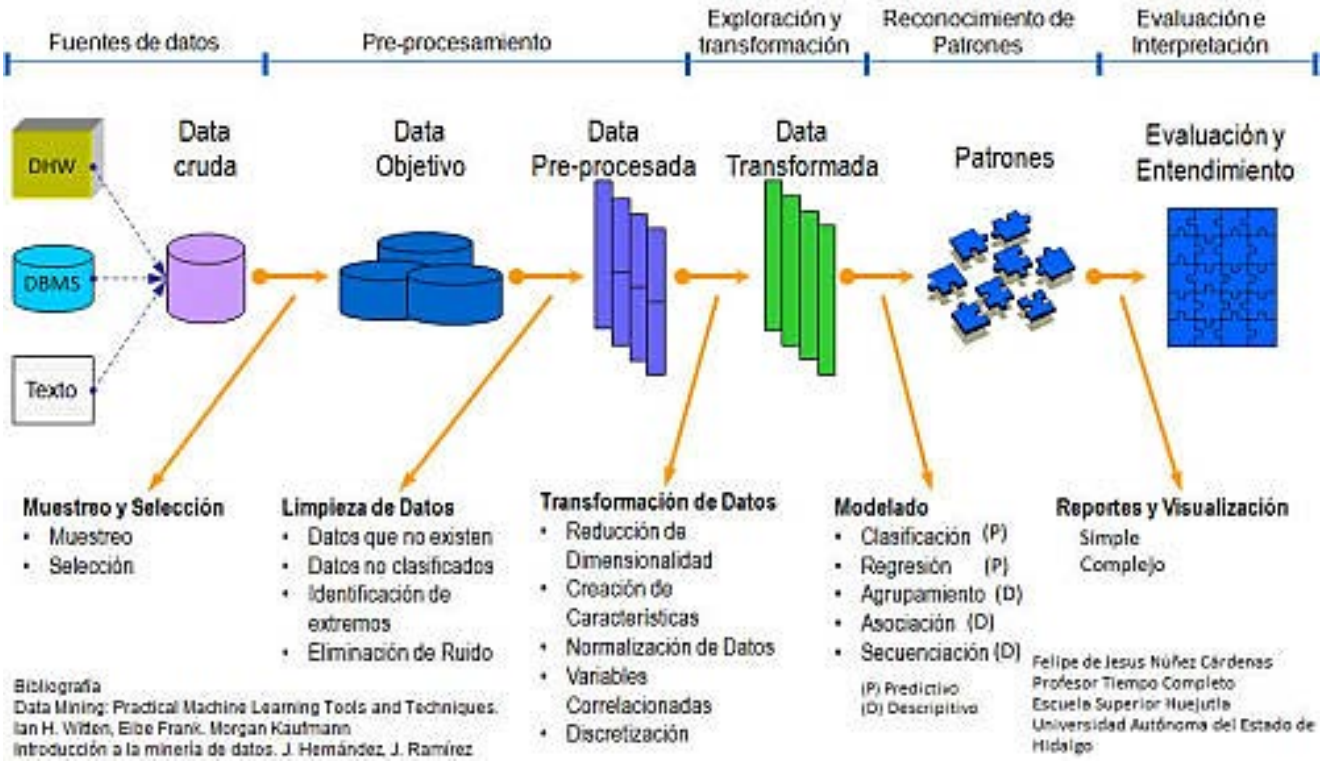


Ilustración 3: El proceso de la minería de datos {6]

- **Selección del conjunto de datos**: En primer lugar se debe elegir una base de datos fiable y suficientemente amplia para resolver el problema. Seguidamente se deciden aquellos datos que se quieren predecir, y posteriormente se seleccionan las variables independientes que se usaran como base de cálculo.
- **Análisis de las propiedades de los datos:** Analizar qué tipo de variables se están obteniendo, y predecir que sesgos pueden proporcionar, ya sea por datos erróneos o datos atípicos o extremos.
- **Transformación del conjunto de datos de entrada**: A partir del análisis anterior, se preparan los datos obtenidos para aplicar la técnica que mejor se adapte al problema. Esto requiere aplicar procesos de limpieza y transformar aquellas variables que por el formato obtenido, no pueden aplicarse correctamente a las técnicas que se usarán. Todo este paso es también conocido como pre-procesamiento de los datos.
- **Selección y aplicación de la técnica de minería de datos:** Se eligen y ejecutan aquellas técnicas que, a priori, optimicen la resolución del problema. Éstas serán detalladas más adelante.
- **Extracción de conocimiento:** Mediante la aplicación de las técnicas anteriores, se obtienen modelos de conocimiento que explican tendencias y patrones en los datos del problema. También se representan relaciones positivas o negativas entre distintas variables.
- **Interpretación y evaluación de datos:** En primer lugar se debe comprobar si las conclusiones que ofrece el modelo son correctas, y en su caso comprobar si resuelven el problema de origen. Si se aplicaron varias técnicas, se debe realizar una comparativa de resultados para comprobar cual resuelve mejor el problema. Si llegados a este punto los resultados no suficientemente apropiados, se debe retroceder a

algún punto anterior, ya sea para cambiar las variables, para modificar la transformación de los datos, o para elegir nuevas técnicas.

Por último es importante realizar una clasificación de las técnicas de Data Mining existentes. La selección de una técnica u otra depende del problema objetivo, o de los tipos de datos a los que se tiene acceso [7]. Se clasifican en:

- **Predictivos**: También denominados métodos de aprendizaje supervisado, tienen por objetivo describir uno o más de los atributos en función del resto. Por tanto, la respuesta al problema se encuentra en los propios datos proporcionados, ya que la aplicación del método escogido sobre ellos proporciona el modelo predictivo. De manera general, los datos se dividen en aquellos de entrenamiento que sirven para generar el modelo, y aquellos de test que ponen a prueba el modelo.

- **Descriptivos**: También denominados métodos de aprendizaje no supervisado, y con ellos, los datos se clasifican en grupos desconocidos anteriormente. Por tanto, con estas técnicas el objetivo es conseguir describir los datos proporcionados, sin conocer posibles vínculos entre ellos a priori.

Una clasificación general de las técnicas se detalla en el siguiente esquema:

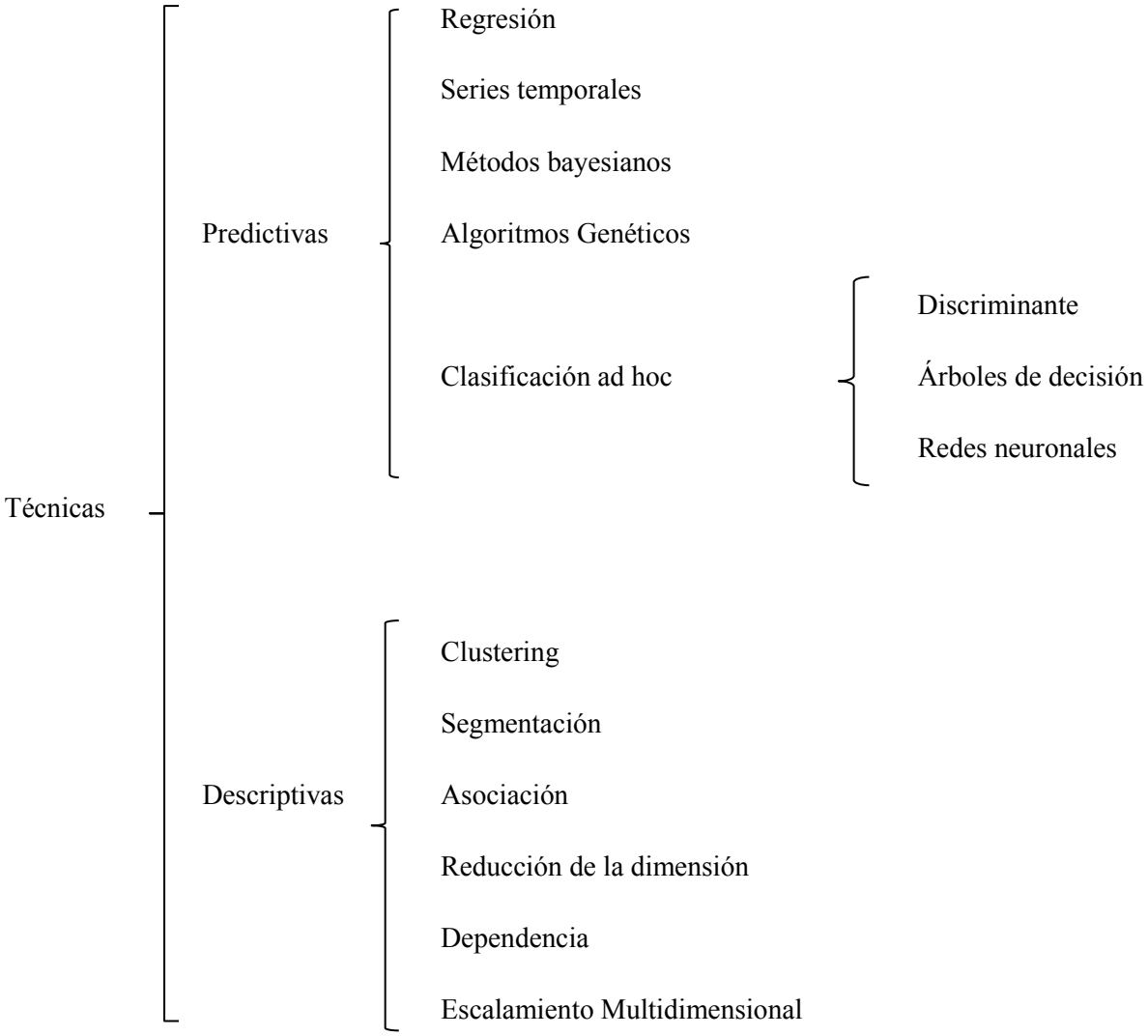


Tabla 2: Técnicas de minería de datos

2.4 Machine Learning (Aprendizaje Automático).

El aprendizaje automático es, resumidamente, el estudio de los algoritmos y programas que permiten mejorar automáticamente basados en la experiencia de acuerdo a alguna medida. Sus objetivos son aprender conocimiento nuevo y mejorar comportamientos. En otras palabras, el aprendizaje automático consiste en mejorar el futuro basándose en las experiencias del pasado.

Por otro lado se debe hacer énfasis en la palabra "automático". El objetivo es concebir algoritmos de aprendizaje que aprendan automáticamente sin intervención humana o asistencia. En lugar de programar al computador para resolver problemas, el aprendizaje automático busca métodos que permitan al ordenador crear sus propios programas basados en ejemplos que nosotros le proporcionamos. Es poco probable que seamos capaces de construir cualquier tipo de sistema inteligente que posea cualquier característica asociada con la inteligencia (como el habla o la visión) sin usar aprendizaje para obtenerlas. Además, no podríamos considerar a un sistema verdaderamente inteligente si fuera incapaz de aprender, ya que el aprendizaje se encuentra en el núcleo de la inteligencia [8].

Algunos ejemplos de problemas en los que se emplea el aprendizaje automático son:

- Reconocimiento óptico: A partir de una imagen ser capaz de clasificarla dentro de un grupo. Esto puede ser aplicado a diferentes problemas como la identificación de rostros.
- Filtros de spam: Identificar correos como deseados o no deseados.

- Categorización de temas: Clasificar un nuevo artículo según su tema, ya sea política, deportes, entretenimiento etc.
- Banca: Como sistema auxiliar de riesgos para la concesión de préstamos.
- Detección de fraudes: Identificar transacciones realizadas mediante una tarjeta bancaria las cuales pueden tener carácter fraudulento.
- Clima: Predicciones relacionadas con la lluvia o la nieve, y sus posibles consecuencias

Anteriormente se realizó una distinción entre aprendizaje supervisado y no supervisado. El objetivo de este proyecto consiste en clasificar instancias, y se espera que los propios datos proporcionados ofrezcan la solución deseada. Es por ello que se emplearán técnicas de aprendizaje supervisado para llegar a la meta.

Dentro del método de aprendizaje supervisado, se van a emplear modelos de clasificación. El efecto de estos métodos es el de ordenar por clases los ejemplos dados.

A continuación se detallan los métodos de clasificación de los que se espera a priori un buen rendimiento.

2.4.1 Métodos de clasificación

• Red de neuronas:

La definición más simple de una red de neuronas artificiales, fue dada por el Dr. Robert Hecht-Nielsen, uno de los primeros neurocomputadores:

"... es un sistema informático formado por un número de elementos simples y altamente interconectados, los cuales procesan información mediante respuestas dinámicas a entradas externas" [9].

Los intentos de construir algoritmos capaces de procesar información al igual que el cerebro humano, han permitido la creación de estas redes artificiales. Por tanto, las redes de neuronas artificiales tienen un evidente fundamento biológico basado en las neuronas naturales.

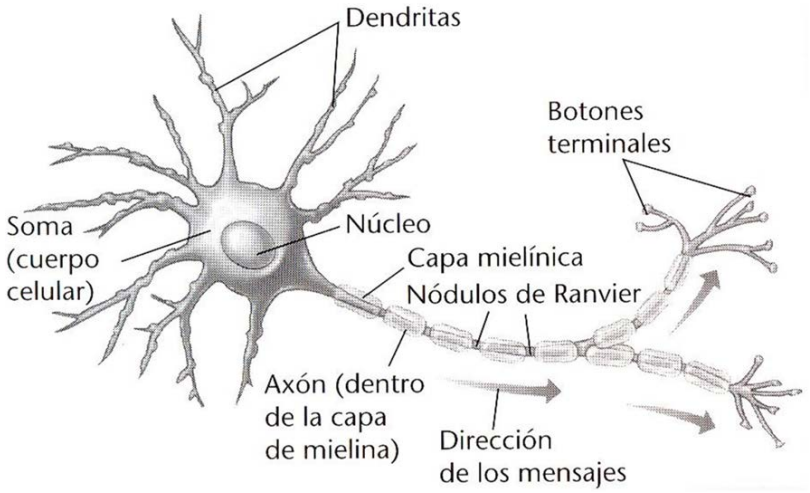


Ilustración 4: Neurona biológica

Una neurona está formada por un cuerpo celular, más conocido como soma. De él, se extienden las dendritas, que sirven como canales de recepción de información. En el soma se procesa la información, y posteriormente, se envía información procesada a través del axón. El trabajo conjunto de una gran cantidad de neuronas es lo que dota de inteligencia a los seres vivos.

Sin embargo, el modelo biológico solo sirve como inspiración, ya que mientras en una red artificial se puede encontrar cientos o miles de unidades, en un cerebro animal se encuentran billones de neuronas.

Una neurona artificial, por su parte, está formada por un conjunto de entradas (x_{ij}), las cuales tienen asignado un peso (w_{ij}). El valor de estos pesos se altera mediante el entrenamiento de la red de neuronas. Por otro lado, en el núcleo de la neurona, se encuentras varios elementos. Cada neurona está caracterizada por un estado interno denominado nivel de activación. Es la función de activación la que permite cambiar el nivel de activación a partir de la

reglas de propagación. Una regla típica puede ser el sumatorio del producto escalar del vector entrada y el vector de los pesos:

$$E = x_1 + w_1 + x_2 + w_2 + \cdots + x_n + w_n + U$$

La función de activación modifica el estado de activación de la neurona si el resultado obtenido con las entradas cumple los requisitos necesarios. La salida generada depende precisamente de ese estado de activación. La siguiente ilustración compara el funcionamiento de una neurona biológica con una artificial.

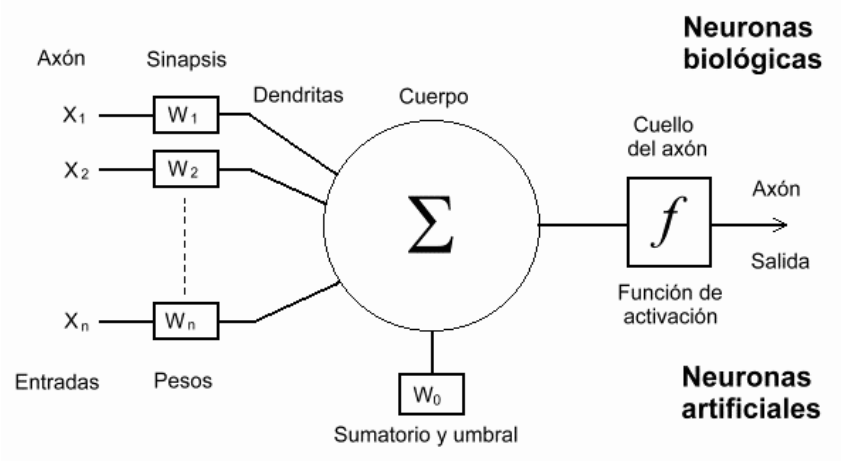


Ilustración 5: Comparación neurona artificial con neurona biológica

Las redes de neuronas artificiales están organizadas típicamente en capas. Éstas están formadas por un número de nodos interconectados, imitando cada uno el funcionamiento de las neuronas recién explicadas. Los patrones entran en la red por la capa de entrada, que se comunican con una o más capas ocultas, donde ocurren las conexiones ponderadas por los pesos. Las capas ocultas se conectan con la capa de salida, que genera el resultado de la red.

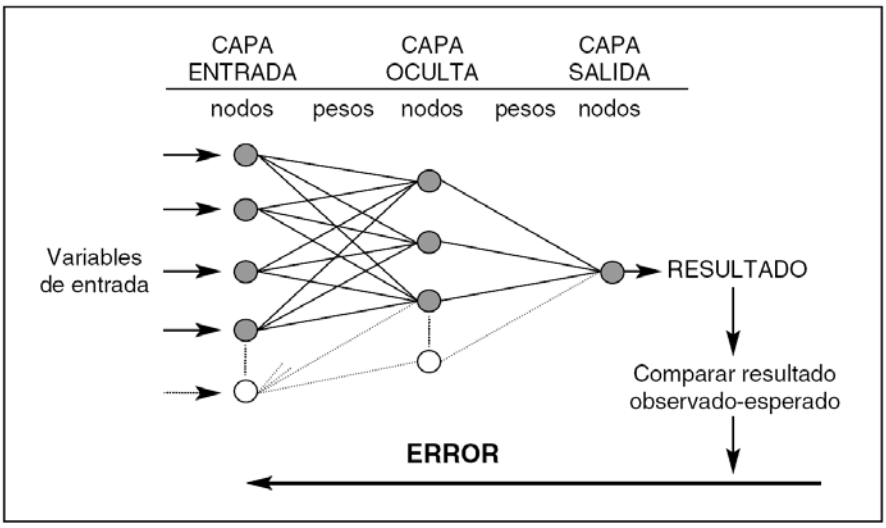


Ilustración 6: Red de neuronas

En este proyecto se utilizará un tipo de red de neuronas conocido como perceptrón multicapa. Estas redes están formadas por múltiples capas, lo que le permite resolver problemas linealmente separables. El proceso de aprendizaje de una perceptrón multicapa, en líneas generales, es el siguiente:

1 - Inicialización de pesos y umbrales aleatorios y próximos a 0.

2 – Presentación de un patrón n de entrenamiento (x(n), s(n)), el cual se propaga a la salida, obteniéndose la respuesta de la red y(n).

3 – Se evalúa el error cuadrático, e(n), cometido por la red para el patrón n.

4 – Se aplican las reglas de modificación de pesos y umbrales de la red.

5 – Se repiten los pasos 2, 3 y 4 hasta llegar al criterio de parada. Este puede ser, alcance de mínimo de error de entrenamiento, estabilidad en el error de entrenamiento, o comienzo de aumento del error de validación.

Es importante comprobar el error de test generado tras varios ciclos. Si este comienza a aumentar, se ha producido sobreaprendizaje, siendo conveniente reducir el número de ciclos de entrenamiento.

- **Árbol de Decisión:**

Los árboles de decisión son representaciones simples para clasificar ejemplos, y es una de las técnicas más eficaces para la clasificación supervisada. El objetivo es crear un modelo que predice el valor de una variable de destino en función de diversas variables de entrada [10].

El siguiente ejemplo es un árbol de decisión muy simplificado de lo que esperamos obtener al finalizar el proyecto. En él se clasifica para un partido si el ganador será el equipo local o el visitante:

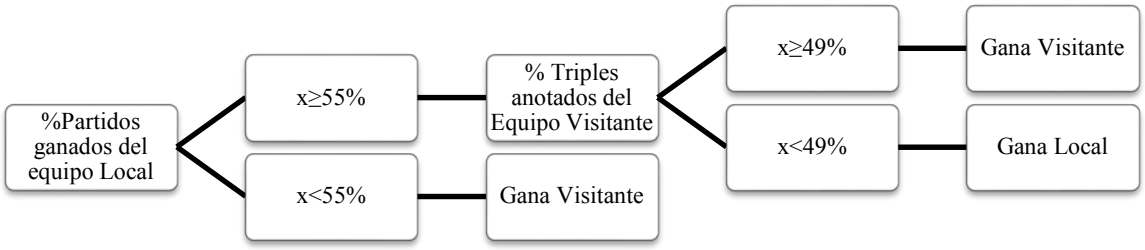


Tabla 3: Ejemplo de árbol de decisión

Cada nodo interior corresponde a una de las variables de entrada. Cada hoja representa un valor de la variable de destino dados los valores de las variables de entrada representados por el camino desde la raíz a la hoja.

El algoritmo empleado será el J48, que es una implementación en Java del algoritmo C4.5 para Weka. [11]

C4.5, creado por Ross Quinlan, se asemeja al algoritmo ID3, pero este incluye post-poda de las ramas del árbol. El algoritmo acepta atributos de entrada booleanos o numéricos, y clases simbólicas, las cuales pueden ser superiores a 2, a diferencia del algoritmo ID3. En definitiva, el algoritmo genera un árbol de decisión usando el concepto de entropía de información. El objetivo es minimizar la entropía, y por tanto, maximizar el orden. Para entender el concepto, es útil pensar en eventos equiprobables. En ellos, la entropía es máxima. Es por ello que el algoritmo se basa en una heurística de ganancia de información, en la que se representa la cantidad que decrece la entropía por elegir uno u otro atributo.

Kotsiantis resume el algoritmo en [12]:

1. Se elige la mejor pregunta para un nodo mediante la heurística de ganancia de información. El objetivo es dividir el conjunto de muestras en subconjuntos que enriquezcan cada clase.
2. Se generan ramas según los valores del atributo.
3. Dividir recursivamente en sub-listas más pequeñas.
4. Respecto a ID3, C4.5 poda el árbol después de la creación. Se remonta a la raíz del árbol eliminando aquellas ramas que no aportan información, sustituyéndolas por nodos hoja.

- **Clasificadores bayesianos:**

Estos clasificadores se basan en el famoso teorema de Bayes, y está basado en la asunción de que las incógnitas de interés sigues distribuciones probabilísticas [13]. La gran aportación de los métodos bayesianos, es que ofrecen una medida probabilística de la importancia de las variables en el problema. Esto diferencia a los clasificadores bayesianos de otros clasificadores como los árboles de decisión o las redes neuronales, los cuales nos dan una medida cuantitativa de esa clasificación [14].

En este proyecto el clasificador bayesiano empleado será el algoritmo Naïve Bayes. La idea del algoritmo es poder estimar las probabilidades a posteriori de cualquier hipótesis consistente con el conjunto de datos de entrenamiento, para así escoger la hipótesis más probable:

Dado un ejemplo x representado por los valores $a_1, \dots a_n$, el algoritmo se basa en encontrar la hipótesis más probable que describa ese ejemplo. Para simplificar el proceso, recurre a la hipótesis de independencia condicional, para poder factorizar la probabilidad. Esta hipótesis enuncia: *Los valores de* a_i *que describen un atributo de un ejemplo son independientes entre sí conocido el valor de la clase a la que pertenecen*. Por tanto, y en definitiva, la probabilidad

de observar la conjunción de atributos a_i dada una categoría a la que pertenecen es el producto de las probabilidades de cada valor por separado [15]:

$$P(a_1, \dots a_n | v_j) = \prod i \; P(a_i | v_j)$$

• **SVM – Máquinas de Soporte Vectorial:**

Las máquinas de soporte vectorial son modelos asociados a la clasificación y a los análisis de regresión. Dado un conjunto de ejemplos de entrenamiento, cada uno marcado por pertenecer a una clase de entre dos, SVM construye un modelo que categoriza nuevos ejemplos en una clase u otra, convirtiéndolo en un clasificador lineal, no probabilístico y binario. Un modelo SVM es una representación de los ejemplos como puntos en el espacio, representados de manera que los ejemplos de cada clase se sitúen a una cierta distancia tan amplia como sea posible. Los nuevos ejemplos son mapeados en el espacio y se predice a que categoría pertenecen según a qué lugar de la diferencia corresponden [16].

Más formalmente, SVM crea un conjunto de hiperplanos en un espacio multidimensional donde se sitúan las diferentes clases. La forma más sencilla de efectuar la separación es mediante una recta, un plano o un hiperplano n-dimensional. Sin embargo, los problemas a tratar no suelen presentar casos de dos dimensiones. Las limitaciones computacionales de las máquinas hacen que la representación por medio de funciones Kernel sea la solución. Los tipos de funciones más utilizados son polinomial-homogénea, perceptron, la sigmoid y la función de base radial gaussiana:

• Polinomial-homogénea: $K(X_i, X_J) = (X_i, X_J)^2$

• Perceptron: $K(X_i, X_J) = ||X_i, X_J||$

• Función de base radial Gaussiana: $K(X_i, X_J) = exp\ (\frac{-K(X_i,X_J)^2}{2(sigma)^2})$

• Sigmoid: $K(X_i, X_J) = tanh(X_i \cdot X_J - \theta)$

Por último, cabe destacar que los modelos SVM ofrecen un entrenamiento muy eficiente, y una buena capacidad de clasificación, funcionando correctamente en problemas típicos. Además, son muy robustos ante la generalización.

• Inducción de Reglas:

Los sistemas de aprendizaje de reglas representan un paradigma transparente, comprensible, aplicable y más genérico que los árboles de clasificación. Muchas de las técnicas empleadas en los sistemas de aprendizaje de reglas han sido adaptadas del aprendizaje mediante árboles de decisión [17].

En este proyecto se empleará el algoritmo de Weka Jrip, que es una implementación java del algoritmo "Repeated Incremental Pruning to Produce Error Reduction (RIPPER)". Este algoritmo fue propuesto por William W. Cogen como una optimización del algoritmo IREP.

El algoritmo JRip es descrito brevemente a continuación [18]:

Se inicializa un conjunto de reglas vacío, y desde la clase menos prevalente a la más frecuente se realizan las siguientes etapas:

1. Etapa de construcción:

   Repetir los pasos 1.1 y 1.2 hasta que no haya ejemplos positivos, o el ratio de error sea mayor al 50%.

1.1 Etapa de crecimiento:

Hacer crecer una regla ávidamente añadiendo antecedentes a la misma hasta que sea perfecta (100% de acierto). El procedimiento intenta cada

posible valor de cada atributo y selecciona la condición con la mayor ganancia de información.

1.2 Etapa de poda:

Podar incrementalmente cada regla. La métrica de poda es (p-n) / (p+n); donde p es el número de ejemplos del subconjunto positivo y n del negativo.

2. Etapa de optimización [19]:

En esta etapa, se ofrecen métricas alternativas para la fase de poda. Se incorporan un heurístico para determinar cuándo parará el proceso de añadir reglas. Posteriormente, se efectúa una búsqueda local para optimizar el conjunto de reglas de dos maneras diferentes: Reemplazando reglas que forman parte del conjunto por otras, siempre y cuando el conjunto de reglas tenga un menor error de clasificación, y añadiendo literales a las reglas para conseguir un menor error.

2.4.2 Conjuntos de Clasificadores

Como afirmó Dietterich en el año 2000 [20], un conjunto de clasificadores, es un grupo de los mismos cuyas decisiones individuales se combinan de alguna manera. Los conjuntos son, frecuentemente, más precisos que un clasificador individual. Para que un conjunto de clasificadores sea más preciso que cualquiera de sus miembros, estos deben ser diversos y también precisos. [21]

Según Dietterich, se pueden encontrar buenos conjuntos de clasificadores por razones estadísticas, computacionales y/o representacionales.

En este proyecto, se emplearan dos métodos de manipulación de los ejemplos de entrenamiento conocidos como Bagging y Boosting. Además, se empleara Random Forest, como método basado en la introducción de aleatoriedad. Por último se incluirá una técnica de Meta-Learning conocida como Stacking, que

incluye el prefijo Meta, ya que consiste en aprender sobre lo aprendido previamente.

- **Bagging**

Bagging, también conocido como bootstrap aggregation, es una técnica que reduce la varianza y ayuda a reducir el sobreajuste. La técnica ayuda a crear diversidad mediante la vuelta a ensamblar de los conjuntos de entrenamiento [22].

La técnica sigue estos pasos [23]:

1. División del conjunto de entrenamiento en T subconjuntos. De esta manera se obtienen T muestras aleatorias con las siguientes características:
    - Misma cantidad de ejemplos en cada muestra.
    - Las muestras se realizan con reemplazo, por tanto, un mismo subconjunto puede contener repetida una instancia.
    - El tamaño de cada subconjunto suele ser igual al tamaño del conjunto de entrenamiento, pero no será igual debido a que los subconjuntos se realizan con reemplazo.
2. Se crea un modelo predictivo con cada set, obteniendo T modelos diferentes.
3. Combina las decisiones de los clasificadores por voto mayoritario.

Es aconsejable que T no sea par, para que no exista empate técnico en la votación. Bagging funciona correctamente cuando más inestables son los modelos. Además, reduce el sobreaprendizaje ya que ningún clasificador tiene todos los datos de entrenamiento.

A continuación se presenta un esquema conceptual de la técnica:

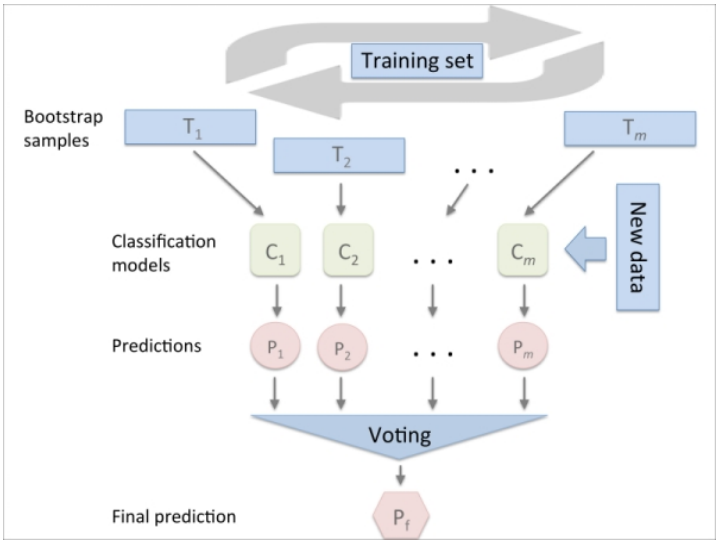


Ilustración 7: Técnica Bagging

- **Random Forest**

Random Forest es un algoritmo que emplea Bagging para combinar diferentes árboles, donde cada uno es construido con variables aleatorias [24]. Antes de profundizar más en la técnica es conveniente resumir el proceso:

1. Elegir ejemplos al azar, con reemplazo, creando así diferentes subconjuntos. En cada subconjunto elegir aleatoriamente un número determinado de atributos.
2. Se crea un árbol de decisión con cada subconjunto, consiguiendo distintos árboles, ya que cada subconjunto contiene diferentes ejemplos y diferentes atributos.
3. Al crear los árboles se escogen atributos aleatoriamente en cada nodo del árbol. De esta manera se deja crecer el árbol en profundidad sin podar.
4. Se predicen los resultados mediante voto mayoritario, clasificando según la mayoría de predicciones de los árboles.

Los principales parámetros que se deben ajustar usando este modelo son el número de estimadores y el número de atributos que tendrá cada subconjunto. El número de estimadores dependerá del número de árboles del bosque. Cuantos más mejor, pero también pondrá en compromiso la capacidad computacional de la máquina. Además, se debe tener en cuenta que los resultados dejarán de ser significativamente mejores a partir de un determinado número de árboles. Por otro lado, los datos empíricos demuestran que el mejor número de atributos a elegir aleatoriamente en cada nodo en los problemas de regresión es la raíz cuadrada del número de atributos [25].

- **Boosting**

Boosting es una técnica empleada para mejorar la precisión de clasificadores débiles. En concreto, en este proyecto se empleará el meta-algoritmo AdaBoost formulado por Yoav Freund y Robert Schapire. La técnica especializa los clasificadores en los ejemplos mal clasificados previamente. Estos clasificadores se generan de forma secuencial y combinan la decisión por votos ponderados.

El primer clasificador, al finalizar su iteración, añade una nueva columna indicando para cada ejemplo si ha sido clasificado correctamente o no. De esta manera, el siguiente clasificador se centrará más en los ejemplos cuyo atributo *error* haya dado positivo en la iteración anterior. La manera de que los clasificadores tengan más en cuenta los ejemplos que tienen más prioridad (es decir, los que han sido mal clasificados más veces en iteraciones anteriores), consiste en clonar los ejemplos tantas veces como malas clasificaciones se haya cometido sobre ese ejemplo previamente.

En la fase de test, las nuevas instancias son evaluadas por todos los clasificadores. Ahora, el método final para clasificar el ejemplo consiste en el voto ponderado. Cada clasificador tendrá un poder de voto inversamente proporcional a su error cometido.

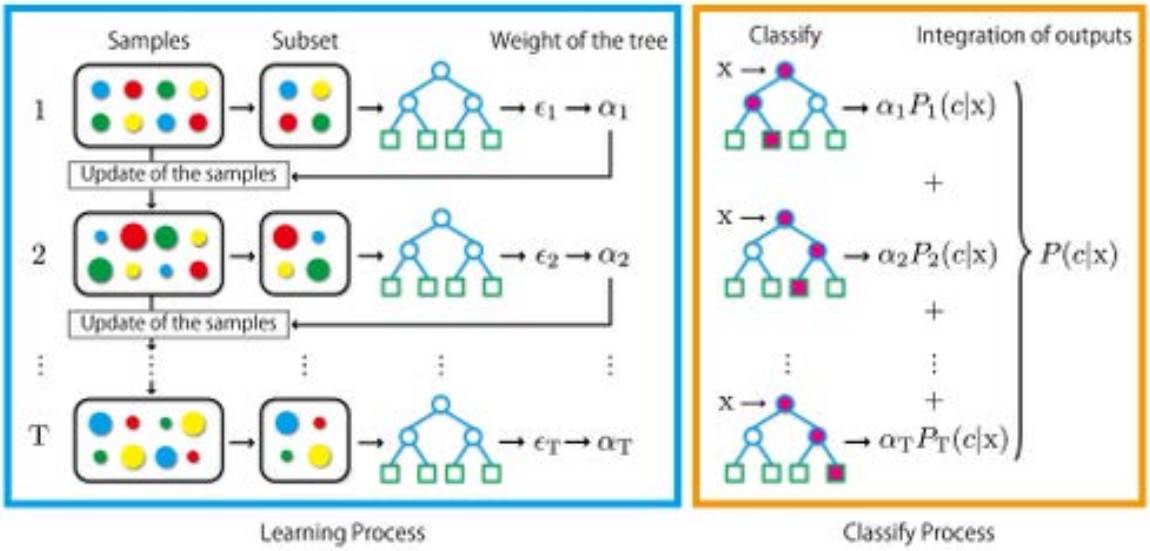


Ilustración 8: Técnica Boosting

- **Stacking**

La técnica genera la clasificación a partir de los resultados obtenidos de varios algoritmos distintos [26]. Por tanto, a diferencia de las técnicas anteriores, en el Stacking si suele observarse como, por ejemplo, una red de neuronas trabaja junto a un árbol de decisión o cualquier otro tipo de técnica de clasificación.

La técnica tiene dos niveles de aprendizaje. El nivel-0 está formado por varios clasificadores, y sus decisiones de combinan en el nivel-1 mediante el concepto de Metaclasificador.

Como se puede en la ilustración 9, entre los niveles 0 y 1 se genera un conjunto de entrenamiento temporal. Este incluye los atributos originales más M_k atributos más, donde k es el número de clasificadores del nivel 0. En estos últimos atributos se almacena la predicción de cada clasificador correspondientemente. En este momento, el Metaclasificador recibe este conjunto de datos temporal, y es el que predice a que clase pertenece cada instancia.

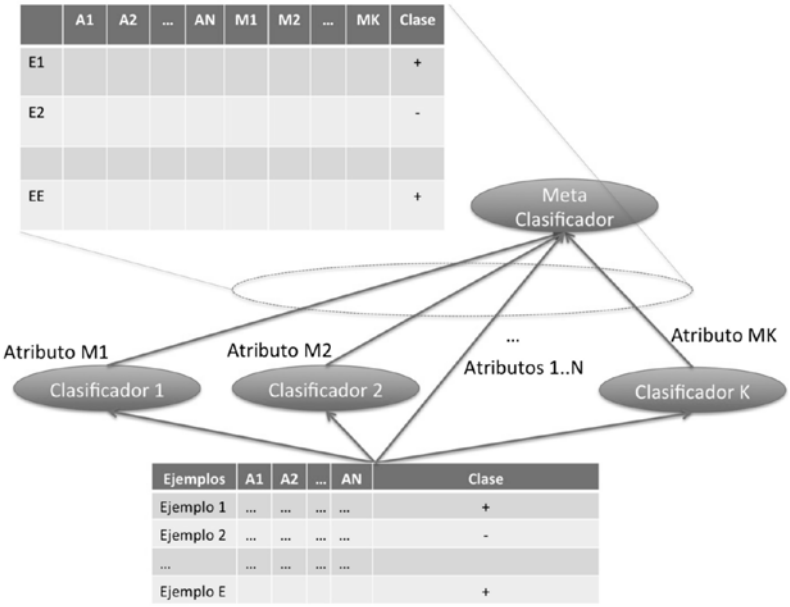


Ilustración 9: Técnica Stacking

2.5 Herramientas y proyectos relacionados con la predicción aplicada al baloncesto

La propia NBA ya ha aplicado la minería de datos en su propia competición. Una de las aplicaciones más conocidas, y sin duda más utilizadas es el "Advanced Scout". Está aplicación, basada en minería de datos, es utilizada por el personal técnico y entrenadores de los equipos para descubrir interesantes patrones a partir de datos y/o imágenes de las grabaciones de los partidos previos. Un ejemplo de su uso nos remonta al 6 de Enero de 1995. New York Knicks se enfrentó a Cleveland Cavaliers, y un análisis de la hoja de jugadas del partido reveló que cuando Mark Price jugaba en la posición de base, John Williams intentó 4 tiros de larga distancia, anotándolos todos. En este caso, la aplicación "Advanced Scout" no solo reconoció este patrón, sino que además recalca la notable diferencia entre el acierto en tiro de este jugador, y el 49,30% que el equipo como conjunto tuvo en ese mismo partido [26].

En materia de predicción de resultados, se han realizado estudios previos que pueden servir como comparador del presente proyecto. Estos estudios son útiles para procurar no cometer los errores descritos en ellos, y para ofrecer mejoras en la predicción mediante atributos más complejos de obtener y nuevos puntos de visto ofrecidos por nuevos algoritmos y la combinación de los mismos. Sin embargo, no es justo comparar los resultados en las tasas de acierto entre ellos, ya que no se está trabajando sobre un mismo dominio de datos. Es decir, mientras este proyecto trabaja sobre la temporada 2015/2016, los demás lo hacen sobre otras anteriores.

El proyecto "Predicting National Basketball Association Winners" elaborado por Jasper Lin, Logan Short y Vishnu Sundaresan [27], utiliza datos estadísticos de la NBA desde el año 1991 al 1998 y desarrolla un modelo de aprendizaje automático para predecir quién ganara un partido determinado.

Para realizar la predicción, únicamente incluyen 17 atributos que son acumulados promedios de las estadísticas obtenidas a lo largo de los partidos, como pueden ser los puntos, los rebotes o las asistencias. Sin embargo, cada instancia se compuso de la diferencia entre los atributos de un equipo frente a los del rival. Una de las conclusiones destacables de este proyecto es, que el porcentaje máximo de acierto para una temporada fue de 65,2% mediante la técnica de Random Forest. Otra conclusión importante, es que los porcentajes de acierto incrementaban notablemente a medida que avanzaba la competición, obteniendo cerca de un 10% más de aciertos al final de temporada respecto al final del primer cuarto de temporada.

Por otro lado, se considera importante el proyecto realizado por Cjenjie Cao del Instituto de Técnología de Dublin mediante técnicas de aprendizaje automático, llamado [28]. A diferencia del proyecto anterior, este incluye 46 atributos que resumen los rendimientos de ambos equipos en los 10 últimos partidos que han disputado antes de enfrentarse el uno al otro. El porcentaje máximo de acierto obtenido es del 69,67% para la temporada 2011.

Además, Bernard Loeffelholz estudió la misma problemática empleando únicamente redes de neuronas [29]. El conjunto de datos empleado en su proyecto consta de los 650 primeros partidos de la temporada 2007/2008, empleando solo los 30 últimos para confeccionar su conjunto de test. El modelo final alcanza el 74,33% de aciertos. Los atributos empleados son las medias de las estadísticas de todos los partidos de los equipos hasta mitad de temporada. El hecho de emplear un conjunto de test tan pequeño y basado simplemente en datos hasta mitad de temporada, dificulta la posibilidad de valorar la practicidad del proyecto.

Como último proyecto de referencia para este experimento, se observa el algoritmo desarrollado por ESPN llamado “Accuscore”, el cual tuvo un porcentaje de acierto de 70,3% en la temporada 2013. [28]

Por último, respecto al análisis predictivo sobre qué equipo anotará primero una canasta en un partido concreto, no existen estudios relacionados conocidos. La única suposición que se puede hacer al respecto la arrojan las cuotas de las casas de apuestas para esa apuesta en concreto, y nos da a entender que la probabilidad de que anote antes un equipo u otro, es a priori, la misma que la de acertar el cara o cruz de un lanzamiento de moneda.

3. Desarrollo

3.1 Metodología

La metodología utilizada en este proyecto está basada en CRISP-DM (Cross Industry Standard Process for Data Mining). En el documento "A data Mining & Knowledge Discovery Process Model [30], se define el modelo como "el estándar de facto para el desarrollo de la minería de datos y los proyectos de descubrimiento de conocimiento". En este apartado van a definirse las diferentes etapas y sus adaptaciones a los objetivos del proyecto.

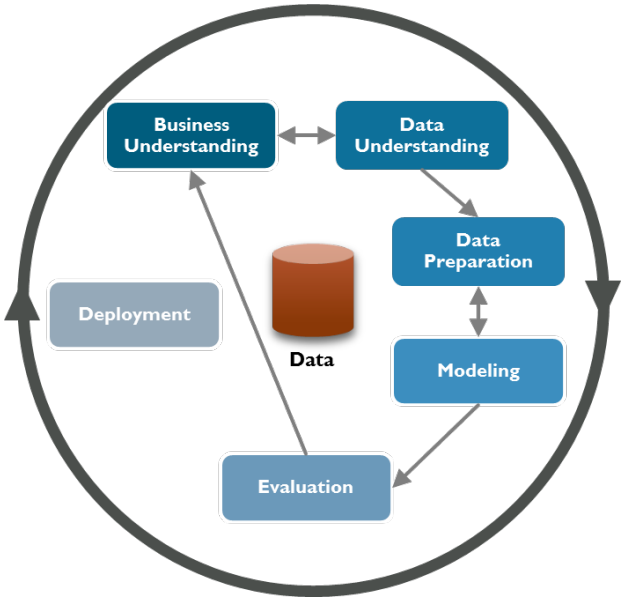


Ilustración 10: Metodología CRISP-DM

- **Comprensión del negocio**

En esta etapa se definen los objetivos de los dos experimentos: Predecir el equipo ganador de un partido de baloncesto, y predecir el equipo que anotará la primera canasta en un partido determinado. Esto supone comprender el entorno del dominio y estudiar los métodos aplicables para lograr los objetivos. En el proyecto, esta etapa se ve reflejada en todos los apartados anteriores.

Además, se confecciona el plan preliminar (apartado 3.1) Y se establecen las herramientas que darán soporte a las siguientes etapas (apartado 3.2).

- **Comprensión de datos**

Esta fase consiste en familiarizarse con los datos del dominio, para poder realizar un nexo entre la comprensión de los objetivos y la preparación de los datos. En este proyecto, la etapa consiste en la consulta de los repositorios de datos para poder intuir que tipo de información se puede extraer y como sería adaptada a las estructuras de las herramientas. Estas estructuras son definidas en el apartado 3.3

- **Preparación de datos**

En esta etapa se cubren todas las actividades para construir el conjunto de datos. Tras las consultas realizadas en los repositorios, se realiza una selección de atributos para ambos experimentos (apartados 3.4.1 y 3.4.2) para posteriormente realizar la extracción de datos (apartado 3.4.3). Después, se emplea un estudio de relevancia de los atributos escogidos para ambos experimentos (apartado 3.5) con el fin de generar los conjuntos de datos finales. Por último, se realiza un pre-procesado al conjunto de datos, para optimizar los conjuntos de entrenamiento y test (apartado 3.6).

- **Modelado**

En esta fase se aplican las técnicas de modelado y se calibran los parámetros para conseguir los mejores resultados posibles. En este proyecto, se emplean las técnicas sobre ambos experimentos en el apartado 4, mostrando los resultados obtenidos técnica por técnica.

- **Evaluación**

En esta etapa se recopilan los datos obtenidos en la fase anterior, de manera que la información pueda ser entendida cómoda y rápidamente. Se comparan los resultados con otros proyectos y se realizan estudios sobre los resultados que puedan exprimir utilidad a todo el proceso anterior. Toda esta etapa se refleja entre los apartados 5.1 a 5.4.

- **Despliegue**

En el apartado 6, se realiza una pequeña explotación del modelado sobre el último 25% de partidos de la temporada regular 2015/2016.

3.2 Herramientas empleadas.

Para el almacenamiento de todos los datos se ha utilizado una hoja de cálculo simple. Entre los motivos por los que se ha decidido emplear esta herramienta, destaca la facilidad para generar filtros. Además, ciertos atributos son calculados a partir de otros, y en las hojas de cálculo resulta sencillo emplear fórmulas que generen rápidamente estos atributos cada vez que se añade una nueva instancia.

En materia de aprendizaje automático, la herramienta básica de este proyecto es Weka.

Ilustración 11: Logo WEKA

Weka está escrito en lenguaje Java y es un software que soporta distintas tareas de minería de datos, y contiene herramientas para el preprocesamiento de datos, clasificación, regresión, clustering, reglas de asociación y visualización [31]. En este proyecto, la herramienta será de gran utilidad para la preparación de los datos, la elección de atributos relevantes, y para la creación de los modelos de predicción.

Para que los conjuntos de datos generados en las hojas de cálculo sean correctamente interpretados por Weka, es necesario formatear los conjuntos en formato CSV al formato Weka (ARFF). Para ello, se emplea un script java que ayuda a automatizar esta parte del proceso.

Por último, se ha usado una herramienta adicional llamada MLP – SPN (Multilayer Perceptron – Simulator Perceptron Network), desarrollada por Antonio de la Mata. Esta aplicación, también escrita en Java, permite configurar una red de neuronas multicapa, para evaluar la evolución del error y encontrar los parámetros que mejor se ajusten al problema.

3.3 Estructura de los ficheros

3.3.1 Estructura del dataset

Como se ha explicado anteriormente, los datos se han recogido en una hoja de cálculo. El objetivo es crear un dataset general con todos los datos y atributos que a priori pueden resultar útiles para los dos experimentos, para posteriormente realizar subconjuntos de forma sencilla.

Ganador	Cuota Local	Cuota Visitante	PorcVictoriasL	PorcVictoriasV	RachaLocal	PPP-L	PPP-Recibidos-L	Tiros-Anotados-L	Tiros-Intentados-L	Tiros--%-L	3P-Anotados L
L	1.83	2.00	0.600	0.600	-1	100.6	102.0	38.0	85.2	0.446	9.8
V	1.55	2.60	0.500	0.250	-1	98.3	98.5	37.0	88.8	0.417	8.5
V	2.30	1.66	0.500	0.500	1	99.0	101.5	36.8	87.8	0.419	8.5
V	2.65	1.54	0.667	0.500	-1	102.7	98.3	34.0	77.7	0.438	3.3
L	2.65	1.54	0.600	0.600	2	101.6	99.6	38.6	83.8	0.461	9.8
L	1.80	2.05	0.250	0.750	-3	100.0	100.8	35.3	87.8	0.402	7.8
V	1.64	2.35	0.000	0.000	-5	89.8	104.4	36.0	82.2	0.438	3.8
L	1.05	12.00	0.800	0.000	4	101.2	90.2	38.6	85.6	0.451	8.6
L	1.01	18.00	1	0.400	5	117.6	96.8	42.4	88.4	0.480	11.2
L	1.95	1.86	0.400	0.600	2	94.4	99.0	34.0	82.6	0.412	8.0
V	2.25	1.68	0.000	0.833	-4	100.8	115.0	37.3	91.0	0.409	9.3
V	1.80	2.05	0.400	0.400	-2	102.0	101.8	37.2	89.0	0.418	7.2
L	2.95	1.42	0.200	1	-1	105.4	106.4	41.4	95.4	0.434	7.6
V	1.54	2.65	0.600	0.750	1	104.0	98.4	38.2	89.0	0.429	8.8

Tabla 4: Fragmento del dataset principal del proyecto

Esta ilustración es solo una pequeña parte (tanto en filas como en columnas) del dataset en conjunto. El formato es simple, se compone de una primera fila o encabezado que da nombre a cada atributo. Cada fila representa una instancia y contiene en cada columna los valores para el atributo que nombra el encabezado.

3.3.2 Estructura de ficheros de entrada a Weka (ARFF)

Una vez generado cada subconjunto de datos a partir de la estructura del apartado anterior, se procede a crear los archivos ARFF. Gracias a que la estructura de estos ficheros es sencilla, con un sencillo script se pueden formatear los archivos CSV a ARFF cómodamente.

Estos ficheros se dividen en tres partes [32]:

1. **"Relation"**

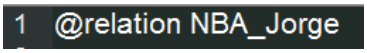


Ilustración 12: Definición de un fichero ARFF

Esta primera línea contiene el nombre del conjunto de datos.

2. **Atributos**

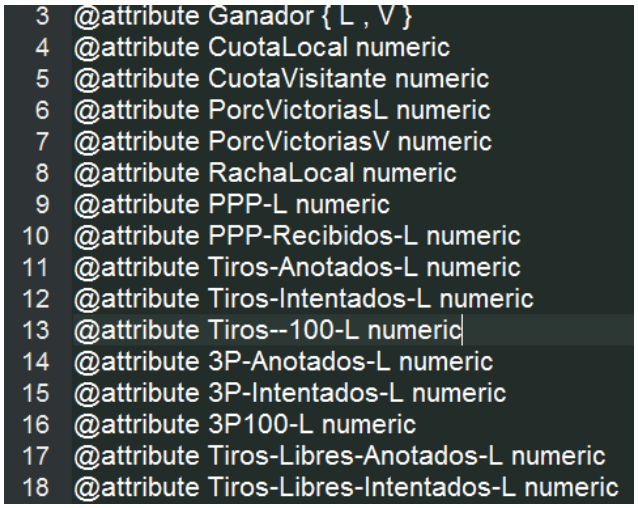


Ilustración 13: Fragmento de definición de atributos de un fichero ARFF

En esta parte del fichero se declaran los atributos. Se debe indicar el nombre, y posteriormente el tipo. En el caso de que el atributo sea discreto, deberá contener el conjunto de valores posibles entre llaves.

3. Datos

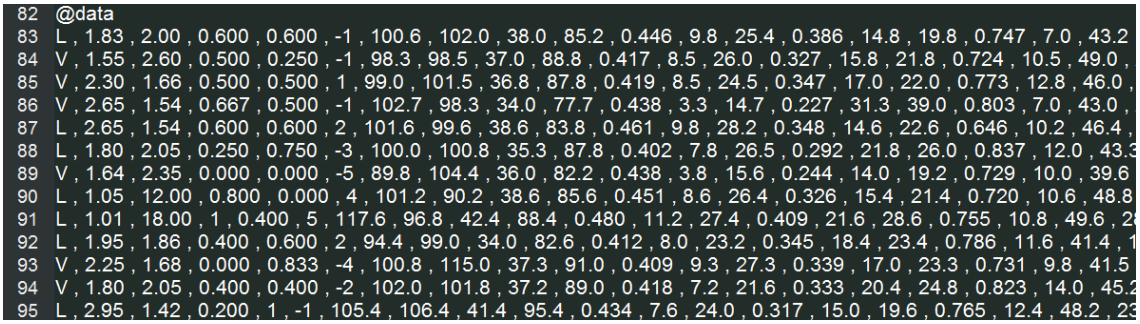


Ilustración 14: Fragmento de definición de instancias en fichero ARFF

La última parte del fichero corresponde a los datos. Cada fila representa una instancia, y los atributos de cada una de ellas es separado por comas.

3.4 Elección de atributos y extracción de datos

En este apartado se van a elegir los atributos que, a priori, pueden ayudar a crear modelos consistentes. Una vez elegidos se procede a la búsqueda de fuentes que puedan contener dichos datos para recopilarlos dándoles un formato útil.

3.4.1 Atributos Escogidos: Ganador.

Antes de explicar qué atributos se han decidido extraer de otras fuentes, es importante recordar y definir el primero de los problemas que se quiere resolver. El objetivo es, a partir de un partido de NBA, clasificarlo o bien como partido en el que gana el equipo LOCAL (L), o partido en el que gana el equipo VISITANTE (V).

Para ello, cada instancia se va a basar en la recopilación de datos agregados de la temporada 2015/2016. En primer lugar se va a definir la clase y

posteriormente se detallará cada atributo para comprender sus posibles valores y qué se espera que aporten al modelo:

• **Ganador:** Este atributo es la clase. Sus posibles valores son: 'L' en caso de que el partido sea ganado por el equipo local, y 'V' en caso de que el partido sea ganado por el equipo visitante.

• **Cuota local y Cuota Visitante:** Se establece un atributo para recoger la cuota local previa al partido, y otro para la cuota visitante.

• **Lesión local y Lesión visitante:** Como se explicó en el apartado 2.1, en la NBA existe un sistema de igualdad, que hace que cada equipo tenga 2 o 3 jugadores destacados, considerados como estrellas, y los cuales son los mejores pagados. Este atributo trata de recoger la inferioridad a la que es sometido un equipo cuando alguno de sus mejores jugadores está lesionado y no puede jugar un partido. Para ello, en primer lugar se ha elaborado una lista de los 5 jugadores mejor pagados de cada uno de los equipos:

ATLANTA	BOSTON	BROOKLYN	CHARLOTTE	CHICAGO	CLEVELAND
Paul Millsap	Amir Johnson	Brook Lopez	Al Jefferson	Derrick Rose	LeBron James
Al Horford	Avery Bradley	Thaddeus Young	Nicolas Batum	Jimmy Butler	Kevin Love
Tiago Splitter	Isaiah Thomas	Jarrett Jack	Kemba Walker	Joakim Noah	Kyrie Irving
Jeff Teague	Jae Crowder	Bojan Bogdanovic	Marvin Williams	Taj Gibson	Tristan Thompson
Kyle Korver	Jonas Jerebko	Sergey Karasev	Michael Kidd-Gilchrist	Pau Gasol	Iman Shumpert

DALLAS	DENVER	DETROIT	GOLDEN STATE	HOUSTON	INDIANA
Wesley Matthews	Danilo Gallinari	Tobias Harris	Klay Thompson	Dwight Howard	Paul George
Chandler Parsons	Kenneth Faried	Reggie Jackson	Draymond Green	James Harden	Monta Ellis
Dirk Nowitzki	Wilson Chandler	Aron Baynes	Andrew Bogut	Corey Brewer	George Hill
Deron Williams	Jameer Nelson	Jodie Meeks	Andre Iguodala	Trevor Ariza	Rodney Stuckey
Zaza Pachulia	Will Barton	Marcus Morris	Stephen Curry	Patrick Beverley	C.J. Miles

LAC	LAL	MEMPHIS	MIAMI	MILWAUKEE	MINNESOTA
Chris Paul	Kobe Bryant	Marc Gasol	Chris Bosh	Greg Monroe	Ricky Rubio
DeAndre Jordan	Roy Hibbert	Zach Randolph	Dwyane Wade	Khris Middleton	Nikola Pekovic
Blake Griffin	Lou Williams	Mike Conley	Goran Dragic	O.J. Mayo	Kevin Garnett
Jeff Green	Nick Young	Lance Stephenson	Luol Deng	Greivis Vasquez	Andrew Wiggins
J.J. Redick	D'Angelo Russell	Brandan Wright	Josh McRoberts	Jabari Parker	Karl-Anthony Towns

NEW ORLEANS	NEW YORK	OKLAHOMA	ORLANDO	PHILADELPHIA	PHOENIX
Eric Gordon	Carmelo Anthony	Kevin Durant	Nikola Vucevic	Carl Landry	Eric Bledsoe
Tyreke Evans	Robin Lopez	Russell Westbroo	Brandon Jennings	Joel Embiid	Tyson Chandler
Omer Asik	Arron Afflalo	Enes Kanter	Ersan Ilyasova	Jahlil Okafor	Brandon Knight
Jrue Holiday	Jose Calderon	Serge Ibaka	Victor Oladipo	Nerlens Noel	Mirza Teletovic
Ryan Anderson	Derrick Williams	Dion Waiters	C.J. Watson	Nik Stauskas	P.J. Tucker

PORTLAND	SACRAMENTO	SAN ANTONIO	TORONTO	UTAH	WASHINGTON
Al-Farouq Aminu	DeMarcus Cousins	LaMarcus Aldridg	DeMarre Carroll	Gordon Hayward	John Wall
Ed Davis	Rudy Gay	Kawhi Leonard	Kyle Lowry	Derrick Favors	Nene Hilario
Gerald Henderson	Rajon Rondo	Tony Parker	DeMar DeRozan	Alec Burks	Marcin Gortat
Chris Kaman	Kosta Koufos	Danny Green	Cory Joseph	Trevor Booker	Markieff Morris
Damian Lillard	Marco Belinelli	Boris Diaw	Patrick Patterson	Dante Exum	Bradley Beal

Tabla 5: Jugadores con mayor salario por equipo

El objetivo es acceder a los reportes previos a cada partido y comprobar que jugadores se pierden el partido. Si, por ejemplo, el equipo local es Washington, y John Wall y Nene Hilario no juegan el partido, el atributo Lesión Local tendrá un valor de 2. Es decir, por cada lesionado del top 5 de salarios de un equipo, el atributo se incrementa en 1.

- **Porcentaje de victorias Local y Visitante:** Estos dos atributos representan el porcentaje de victorias de los equipos hasta la fecha del partido. Se espera que ofrezcan una medición del rendimiento global del equipo.

- **Racha Local y Racha Visitante:** El objetivo de la inclusión de estos dos atributos es medir el rendimiento del equipo en los últimos partidos. Si un equipo tiene una racha negativo, por ejemplo, de 3 partidos seguidos perdidos, el valor del atributo será "-3". Si un equipo ha ganado 4 partidos seguidos, su valor será "4".

- **Racha Local contra el Visitante y Racha Visitante Contra el Local:** Mide el número de partidos seguidos ganados o perdidos que ha tenido el equipo local contra el visitante y viceversa.

• **Puntos por partido Local y Visitante:** Estos dos atributos muestran la media de puntos de cada equipo durante la temporada hasta el día antes del partido. Se espera que ofrezcan un indicador global de la capacidad de ataque de los equipos.

• **Puntos encajados por partido Local y Visitante:** Estos dos atributos muestran la media de puntos que suele recibir en contra cada equipo durante la temporada. Se espera que ofrezcan un indicador global de la capacidad defensiva de los equipos.

• **Tiros de 2 intentados, Tiros de 2 Logrados, Porcentaje acierto tiros de 2 (Para Local y Visitante):** Representan el potencial ofensivo en tiros de 2 puntos. Se momento, se incluyen los dos primeros atributos para estimar en qué medida los equipos rivales evitan que el equipo lance.

• **Tiros de 3 intentados, Tiros de 3 Logrados, Porcentaje acierto tiros de 3 (Para Local y Visitante):** Misma función que el atributo anterior pero con los tiros de larga distancia.

• **Tiros libres intentados, Tiros libres Logrados, Porcentaje acierto tiros libres (Para Local y Visitante):** Misma función que el atributo anterior pero con los tiros libres.

• **Promedio de Rebotes Ofensivos Local y Visitante:** Representa la capacidad de cada equipo para generar nuevas jugadas en ataque.

• **Promedio de Rebotes Totales Local y Visitante:** Representa la capacidad de cada equipo de generar nuevas jugadas en general.

• **Promedio de Asistencias Local y Visitante:** Representa para cada equipo el promedio de asistencias, es decir, mide el rendimiento ofensivo.

• **Promedio de Robos Local y Visitante:** Representa para cada equipo el promedio de robos, es decir, mide el rendimiento defensivo.

• **Promedio de Pérdidas Local y Visitante:** Representa para cada equipo el promedio de pérdidas.

• **Promedio de Faltas Personales Local y Visitante:** Representa para cada equipo el promedio de faltas cometidas.

• **Tiros de 2 intentados por los oponentes, Tiros de 2 Logrados por los oponentes, Porcentaje acierto tiros de 2 de los oponentes (Para Local y Visitante):** Estos atributos miden el potencial ofensivo que los equipos permiten tener a sus rivales. Es decir, miden el rendimiento en tiros de 2 que han tenido los oponentes de un equipo en los partidos contra ellos durante la temporada.

• **Tiros de 3 intentados por los oponentes, Tiros de 3 Logrados por los oponentes, Porcentaje acierto tiros de 3 de los oponentes (Para Local y Visitante):** Misma función que el atributo anterior pero con los tiros de larga distancia.

• **Tiros de 2 intentados por los oponentes, Tiros de 2 Logrados por los oponentes, Porcentaje acierto tiros de 2 de los oponentes (Para Local y Visitante):** Misma función que el atributo anterior pero con los tiros libres.

• **Promedio de Rebotes Ofensivos de los oponentes - Local y Visitante:** Representa la capacidad de los contrincantes de un equipo para generar nuevas jugadas en ataque contra ellos.

• **Promedio de Rebotes Totales de los oponentes Local y Visitante:** Representa la capacidad de los contrincantes de un equipo para conseguir rebotes contra ellos.

• **Promedio de Asistencias de los oponentes Local y Visitante:** Representa la capacidad de los contrincantes de un equipo para hacer asistencias contra ellos.

• **Promedio de Robos de los oponentes Local y Visitante:** Representa la capacidad de los contrincantes para robarles la posesión.

• **Promedio de Pérdidas de los oponentes Local y Visitante:** Mide el promedio de veces que los contrincantes pierden la posesión contra ellos

• **Promedio de Faltas Personales de los oponentes Local y Visitante:** Representa para cada equipo el promedio de faltas recibidas.

• **Días libres Local y Visitante:** Representan los días de descanso que ha tenido cada equipo los días previos al partido. El objetivo es comprobar si los equipos con más días de descanso tienen una ventaja notable.

3.4.2. Atributos Escogidos: Primer equipo en anotar.

El objetivo de este segundo experimento es, a partir de un partido de NBA, predecir si anotará primero el equipo Local (L) o Visitante (V). Para ello los atributos escogidos son:

• **Primer Canasta:** Es la clase. Toma el valor L si la primera canasta la anota el equipo local y V si la primera canasta la anota el equipo visitante.

• **Porcentaje Primera Posesión Local:** Muestra el porcentaje de partidos en los que el equipo local comenzó ganando la primera posesión tras el saque neutral. El objetivo es comprobar si los equipos que suelen tener la primera posesión más a menudo suelen ser los primeros en anotar.

• **Porcentaje Primera Posesión Visitante:** Mismo objetivo que el atributo anterior pero muestra el porcentaje del equipo visitante.

• **Porcentaje Primer Tiro Local:** Muestra el porcentaje de partidos en los que el equipo local anota su primer tiro, independientemente de si ha sido el primer equipo en anotar. El objetivo es comprobar si los equipos que anotan con frecuencia su primer lanzamiento, suelen ser el primer equipo en anotar.

• **Porcentaje Primer Tiro Visitante:** Mismo objetivo que el atributo anterior pero muestra el porcentaje del equipo visitante.

• **Porcentaje Primer Tiro Anotado:** Muestra el porcentaje de partidos en los que el equipo local es el primer equipo en anotar. El objetivo es comprobar si los equipos que son los primeros en anotar con frecuencia, tienen más probabilidad de serlo en partidos futuros.

Los experimentos se realizarán con estos atributos, pero también, se incluirán pruebas añadiendo los atributos explicados en el apartado 3.4.1 para comprobar si estos tienen relevancia en este problema. Es decir, se comprobará si los atributos que predicen qué equipo ganara un partido, pueden ayudar a predecir qué equipo anotará primero.

3.4.3 Extracción de datos

El problema de esta parte del proyecto es que el origen de los datos no permite una fase totalmente automatizada, y por tanto la gran mayoría de los datos han sido recogidos de forma manual. Esto ha sido causado por las características de los atributos. Los orígenes de los datos son muy dispares, y las consultas han tenido que ser realizadas en diferentes zonas del portal, dificultando la creación de un proceso automático.

En especial, el proceso manual más complejo ha sido la recopilación de datos para el problema de predicción del primer equipo en anotar. El proceso consiste en recurrir al reporte de cada partido, acceder al listado de jugadas y anotar lo ocurrido para establecer los valores de cada atributo. Para calcular los porcentajes explicados en el apartado 3.4.2, se ha creado una tabla Excel para calcular el valor agregado en cada momento de la temporada. Esta tabla se puede consultar en el anexo 2.

Los datos han sido recogidos de dos fuentes distintas. Para obtener el histórico de cuotas se ha recurrido a Oddsportal [33]. El resto de datos se ha

obtenido del portal Basketball Reference [34]. Esta web no solo proporciona un repositorio de datos fiable y conectado a la NBA. También ofrece una cantidad muy alta de datos detallados y, en ocasiones, calculados a raíz de la recopilación de datos en bruto. Una de las herramientas que ofrece es la de la consulta de datos agregados desde el comienzo de la temporada hasta una fecha en concreto. Esto permite, y facilita la recopilación de aquellos atributos que representan el rendimiento agregado del equipo exactamente en el día previo a una nueva instancia.

3.5 Relevancia de atributos – Atributos finales

Para estudiar la correlación de los atributos con la clase, se van a utilizar dos algoritmos implementados en Weka:

- **Chi Squared Attribute Evaluation:** El test chi cuadrado es utilizado en estadística y otros campos para probar la independencia de dos eventos. Específicamente, en la selección de atributos, el test es utilizado para comprobar si la ocurrencia de un atributo es independiente de la clase. Valores altos en el test chi cuadrado indican que la hipótesis nula de la independencia debe ser rechazada, y por tanto, el atributo y la clase son dependientes [35].

  El problema que presenta este test, es que, debido al grado de libertad, una pequeña parte de los atributos seleccionados sea independiente de la clase. Sin embargo, estos atributos ruidosos no afectan seriamente a la precisión de los clasificadores [36].

- **Gain Ratio Attribute Evaluation:** Este test evalúa el valor de un atributo midiendo su ratio de ganancia respecto a la clase. La evaluación mediante Gain Ratio se basa en el concepto de entropía, y su medida nos

aporta en qué grado un atributo aporta información para resolver el problema.

Una vez realizadas ambas evaluaciones en los distintos problemas, los atributos elegidos serán aquellos que superen el valor 0 en ambas evaluaciones.

3.5.1 Atributos Relevantes: Ganador

A continuación se muestran las puntuaciones en ambas pruebas para cada atributo. Las letras L y V indican si el atributo corresponde al equipo local o al visitante. Por otro lado, los atributos que incluyan el término OP, de oponente, hacen referencia a que ese dato corresponde a datos encajados por un equipo. Por ejemplo, el atributo "Rebotes Totales L OP" recoge el promedio de rebotes por partido que han tenido los equipos que han jugado contra el equipo local a lo largo de la temporada. Es decir, un valor alto en este atributo implica que los equipos que juegan contra este equipo (que juega de local) tienen facilidad para capturar rebotes contra él.

Atributo	Test Chi Square	Test Gain Ratio
Cuota Visitante	244,11765	0,0774
Cuota Local	240,22246	0,0724
Porc. Victorias L	88,45034	0,0502
Porc. Victorias V	87,38121	0,0381
Porc. Tiros L OP	54,64725	0,0358
Puntos Por Partido V	50,77163	0,0302
Triples Intentados L OP	50,55388	0,0321
Puntos Recibidos Por Partido L	50,17949	0,0286
Puntos Por Partido L	49,15564	0,0293
Porc. Tiros L	49,10119	0,0726
Asistencias L OP	45,99982	0,029
Porc. Tiros V OP	44,73097	0,0281
Tiros Anotados L OP	41,51591	0,026
Tiros Anotados V	39,05373	0,0484
Porc. Triples L	38,62602	0,0247
Puntos Recibidos Por Partido V	37,25161	0,0233
Asistencias V OP	37,13952	0,0232
Asistencias L	37,04918	0,0578
Porc. Tiros V	36,19667	0,0489
Triples Anotados L	35,90695	0,0469
Rebotes Ofensivos L OP	35,72441	0,0143
Triples Anotados L OP	33,72908	0,0321

Tiros Anotados L	33,2273	0,0549
Porc. Triples V	31,73148	0,0828
Rebotes Totales L	31,69675	0,0231
Rebotes Totales L OP	30,64306	0,0318
Triples Anotados V	30,3556	0,0888
Tapones L OP	29,79615	0,0417
Tapones V OP	29,37938	0,0191
Triples Anotados V OP	28,96369	0,0779
Asistencias V	28,94212	0,0861
Rebotes Totales V	24,78631	0,0155
Tiros Anotados V OP	23,3422	0,0146
Porc. Triples L OP	22,44517	0,0301
Pérdidas L	22,08019	0,0329
Triples Intentados V	22,00452	0,0291
Triples Intentados V OP	20,25212	0,0126
Triples Intentados L	19,56174	0,0147
Rebotes Totales V OP	19,15402	0,0151
Tiros Intentados L OP	18,54014	0,0175
Porc. Triples V OP	17,49824	0,0113
Tiros Libres Anotados L OP	17,37856	0,109
Faltas Personales V OP	14,99608	0,011
Porc. Tiros Libres V OP	13,13831	0,1028
Racha Visitante	12,54776	0,0271
Faltas Personales L	11,73478	0,1004
Tapones L	0	0
Robos L	0	0
Racha Local	0	0
Tiros Intentados L	0	0
Tiros Libres Anotados L	0	0
Tiros Libres 100 L	0	0
Rebotes Ofensivos L	0	0
Tiros Libres Intentados L	0	0
Días Libres V	0	0
Tiros Libres Intentados L OP	0	0
Tiros-Intentados V OP	0	0
Faltas Personales V	0	0
Tapones V	0	0
Tiros Libres Anotados V OP	0	0
Tiros Libres Intentados V OP	0	0
Rebotes Ofensivos V OP	0	0
Robos V OP	0	0
Perdidas V OP	0	0
Perdidas V	0	0
Robos V	0	0
Porc. Tiros Libres L OP	0	0
Días Libres L	0	0
Robos L OP	0	0
Rebotes Ofensivos V	0	0
Faltas Personales L OP	0	0
Tiros Intentados V	0	0
Tiros Libres Anotados V	0	0
Tiros Libres Intentados V	0	0
Porc. Tiros Libres V	0	0
Perdidas L OP	0	0

Tabla 6: Estudio de relevancia de atributos

Un primer análisis nos permite comprobar que, como se predijo, los atributos Cuota Local, Cuota Visitante y los porcentajes de victorias de ambos equipos,

son los atributos con mayor correlación con la clase. Todos los atributos con una puntuación de 0 en ambas pruebas serán excluidos.

3.5.2 Atributos Relevantes: Primer equipo en Anotar.

La prueba de relevancia de atributos para el problema de predicción de qué equipo será el primer en anotar se ha realizado con 302 instancias, correspondientes a los primeros 302 partidos de la temporada 2015/2016. Esto es así debido a la complejidad temporal de la extracción de los valores para los atributos empleados en este problema. Con estas primeras 302 instancias se ha querido estudiar si los atributos seleccionados empiezan a tener relevancia, para, en caso positivo, seguir recopilando datos.

Se ha de recordar que para este problema, primero se estudiará el pequeño conjunto de atributos explicado en el apartado 3.4.3, referentes al rendimiento de los equipos a la hora de obtener la primera posesión, su habilidad para anotar la primera canasta, y el porcentaje de partidos en los que suelen ser los primeros en anotar. Posteriormente se incluirán los atributos referentes al problema de predicción de equipo ganador para observar si el hecho de ser favorito para ganar un partido, condiciona la probabilidad de ser el primer equipo en anotar.

Al emplear los test de Chi Cuadrado y Gain Ratio, se observa que el resultado para el primer conjunto de datos no es bueno. Todos los atributos tienen una puntuación de 0 en ambas pruebas. Por tanto, Ningún atributo parece tener correlación con la clase, por lo que presumiblemente el porcentaje de acierto, en un escenario optimista rondará el 55%.

Si incluimos en el test los atributos de problema de predicción de equipo ganador, todos los atributos vuelven a tener una puntuación de 0, por lo que de nuevo, la previsión de acierto no es muy buena. Sin embargo, la

experimentación se llevará a cabo para confirmar los bajos porcentajes de acierto en este problema.

3.6 Pre-Procesamiento de datos

Antes de comenzar la experimentación es necesario aplicar técnicas de pre-procesamiento de datos para obtener mejores resultados. Al conjunto de datos final se le aplicarán los siguientes filtros:

- **Borrado de instancias poco relevantes:** En una temporada de NBA se juegan un total de 1230 partidos que comienzan el 27 de octubre. Sin embargo, en las primeras jornadas los datos acumulados de los rendimientos de los equipos a lo largo de la campaña son insuficientes o irrelevantes. Por eso se ha decidido podar las primeras jornadas, reduciendo el número de instancias a 1161. Se considera que a partir del 5 de noviembre de 2015 se tienen datos con cierta relevancia para todos los equipos. En definitiva, se han suprimido los primeros 69 partidos de la temporada.
- **Borrado de atributos irrelevantes:** Todos aquellos atributos con una puntuación de 0 en los dos test de relevancia realizados, serán eliminados de todas las instancias.
- **Normalización:** Este proceso estandariza los rangos de los atributos de un conjunto de datos. En algunos algoritmos, si no se aplica un proceso de normalización, los valores extremos y la gran amplitud de los rangos pueden provocar que los resultados no sean suficientemente correctos. Por ejemplo, si un algoritmo calcula la distancia euclídea entre dos puntos sin los datos normalizados, aquellos atributos con rangos amplios de valores tomarán mayor relevancia en el clasificador. Por ello, el rango de todos los atributos debe ser normalizado para

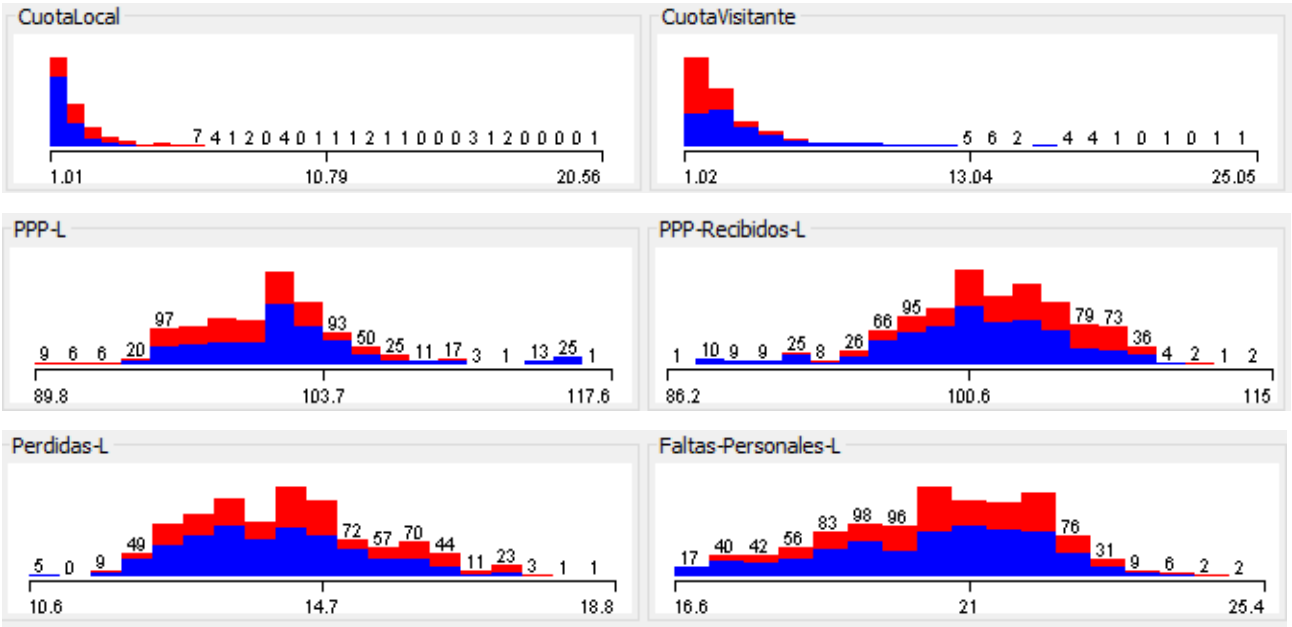
que todos ellos contribuyan aproximadamente de manera proporcional a la distancia final [37].

En este proyecto se va a realizar una normalización de todos los datos numéricos en el rango [0,1], mediante la siguiente formula:

$$V' = \frac{\mathrm{V} - \min(\mathrm{V})}{\max(V) - \min(V)}$$

Donde V es un atributo original y V' el valor normalizado.

Los atributos que representan un porcentaje medio de la temporada ya tienen un rango de 0 a 1, sin embargo, es importante asegurarse que otros valores tienen rangos muy dispares. Algunos de estos atributos se muestran a continuación, representando el eje X el rango de valores, y los colores azul (L) y rojo (V) las distribuciones absolutas para cada clase:



Gráfica 1: Distribuciones absolutas para ciertos atributos

Simplemente con la visualización de los rangos de estos 6 atributos se comprueba la disparidad de valores, y la necesidad de realizar una normalización de los datos.

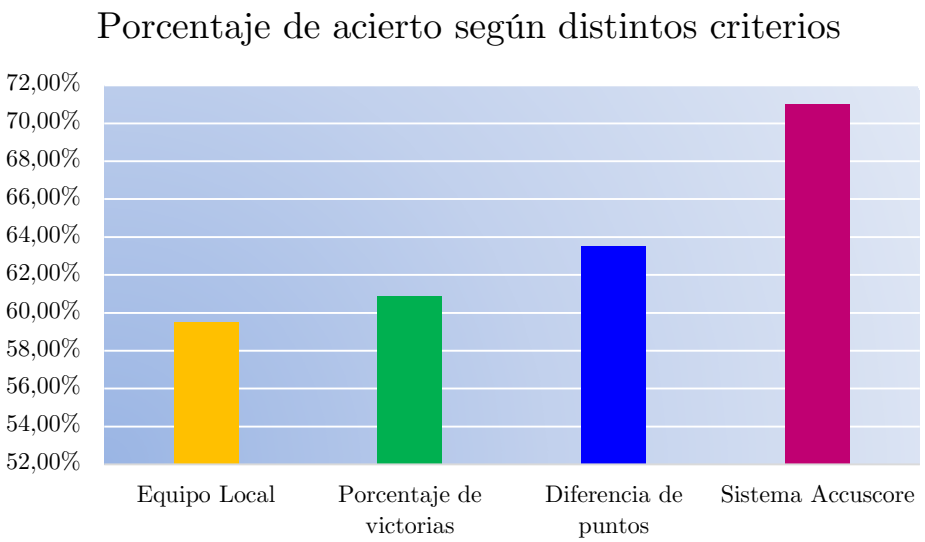
4 Experimentación

En este apartado se van a mostrar los resultados obtenidos mediante la aplicación de las distintas técnicas de aprendizaje automático descritas. Se mostrarán los resultados tal y como han sido obtenidos, para posteriormente realizar comparativas y análisis en el apartado 5.

4.1 Experimentación previa

Antes de recurrir a las técnicas de aprendizaje supervisado, resulta interesante observar los datos y realizar pequeñas comprobaciones estadísticas con los datos extraídos.

En primer lugar, se observa que en el 59,51% de los partidos, el equipo ganador es el equipo local. Por otro lado, en un 60,9% de los partidos, el equipo ganador es aquel con mayor porcentaje de victorias durante la temporada, y en un 63,5% de ocasiones el equipo ganador es el que tiene una mayor diferencia entre su promedio de puntos anotados por partido y su promedio de puntos recibidos por partido. Por último, y a modo comparativo, se debe recordar que el sistema Accuscore posee un porcentaje de acierto del 70,3%.



Gráfica 2: Porcentajes de acierto según distintos criterios

Comparar los resultados obtenidos aplicando técnicas de aprendizaje automático con esta experimentación previa, puede arrojar interesantes conclusiones. El porcentaje de victorias y la diferencia de puntos son indicadores globales del rendimiento del equipo. Si en la experimentación que se va a llevar a cabo se obtienen porcentajes de acierto más altos, será debido a que la inclusión de más atributos nos ayuda a capturar y evaluar ciertas habilidades del equipo que no son recogidas en estos indicadores globales, como la capacidad ofensiva o defensiva de un equipo.

4.2 Experimentación: Ganador

Este apartado muestra la aplicación de las técnicas de aprendizaje supervisado para predecir qué equipo ganará un partido determinado.

Es importante destacar que para todas las técnicas el conjunto de datos va a dividirse de la siguiente manera: el primer 75% de los partidos conformarán el set de datos para entrenamiento, mientras que el último 25% de partidos de la temporada formarán el conjunto de test. Es decir, el conjunto de entrenamiento está formado por los primeros 871 partidos, y el conjunto de test por los 290 últimos.

Los resultados para cada técnica son los siguientes:

• **Naïve Bayes:** Para este clasificador se activa el estimador kernel para los atributos numéricos en lugar del estimador de distribución normal.

```
Correctly Classified Instances         209               72.069  %
Incorrectly Classified Instances        81               27.931  %
Kappa statistic                          0.4265
Mean absolute error                      0.306
Root mean squared error                  0.4964
Relative absolute error                 63.7285 %
Root relative squared error            101.6978 %
Total Number of Instances              290

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.729     0.292      0.796     0.729     0.761      0.77     L
                 0.708     0.271      0.625     0.708     0.664      0.77     V
Weighted Avg.    0.721     0.284      0.73      0.721     0.723      0.77

=== Confusion Matrix ===

   a   b   <-- classified as
 129  48 |   a = L
  33  80 |   b = V
```

Ilustración 15: Resultados Naïve Bayes - Experimento 1

Como se puede comprobar, de los 290 partidos del test de entrenamiento se clasifican correctamente 209 (72,06% de acierto).

• **SVM:** Para esta técnica se va a emplear una función de kernel polinomial homogénea. El parámetro de coste se configuró en 4 y se activó la probabilidad estimada.

```
Correctly Classified Instances         202               69.6552 %
Incorrectly Classified Instances        88               30.3448 %
Kappa statistic                          0.3137
Mean absolute error                      0.3034
Root mean squared error                  0.5509
Relative absolute error                 63.1979 %
Root relative squared error            112.8584 %
Total Number of Instances              290

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.876     0.584      0.701     0.876     0.779      0.646    L
                 0.416     0.124      0.681     0.416     0.516      0.646    V
Weighted Avg.    0.697     0.405      0.693     0.697     0.677      0.646

=== Confusion Matrix ===

   a   b   <-- classified as
 155  22 |   a = L
  66  47 |   b = V
```

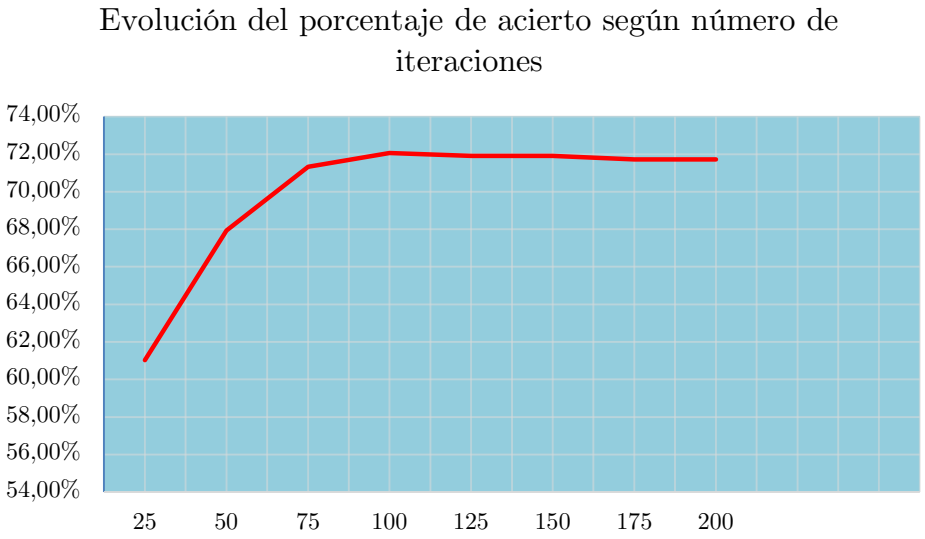
Ilustración 16: Resultados SVM - Experimento 1

Esta técnica clasificó correctamente 202 partidos (69,65%).

- **Perceptrón Multicapa:**

En primer lugar, mediante el programa SPN se ha buscado la configuración óptima. Tras numerosas pruebas, se han encontrado valores que estabilizan el error. Estos son una tasa de aprendizaje del 0,005 y la inclusión de una capa oculta de 20 neuronas. Con esta configuración, el error decrece lentamente entre las iteraciones 100 y 200 en adelante.

A continuación, se configura la red de neuronas en Weka con los datos óptimos obtenidos en SPN. Ahora, para evitar la sobreadecuación, se calcula la evolución de la tasa de acierto según el número de iteraciones empleadas.



Gráfica 3: Evolución porcentaje de acierto en Red de Neuronas

La tasa de acierto más alta se obtiene con 100 iteraciones, siendo esta del 72,06%. A partir de ahí, decrece hasta el 71,72%, manteniéndose constante durante varias iteraciones.

```
Correctly Classified Instances         209               72.069  %
Incorrectly Classified Instances        81               27.931  %
Kappa statistic                          0.3821
Mean absolute error                      0.3987
Root mean squared error                  0.4333
Relative absolute error                 83.0425 %
Root relative squared error             88.7772 %
Total Number of Instances              290

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.859     0.496      0.731     0.859      0.79       0.786    L
                 0.504     0.141      0.695     0.504      0.585      0.786    V
Weighted Avg.    0.721     0.358      0.717     0.721      0.71       0.786

=== Confusion Matrix ===

   a   b   <-- classified as
 152  25 |   a = L
  56  57 |   b = V
```

Ilustración 17: Resultados Red de Neuronas - Experimento 1

• **Árbol J48:** Para generar el árbol de decisión se ha configurado el factor de confianza a 0.1 para que el sistema de poda reduzca el tamaño del árbol, y el número mínimo de instancias por hoja a 2.

```
Correctly Classified Instances         211               72.7586 %
Incorrectly Classified Instances        79               27.2414 %
Kappa statistic                          0.3994
Mean absolute error                      0.4062
Root mean squared error                  0.4457
Relative absolute error                 84.5911 %
Root relative squared error             91.3103 %
Total Number of Instances              290

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.859     0.478      0.738     0.859      0.794      0.69     L
                 0.522     0.141      0.702     0.522      0.599      0.69     V
Weighted Avg.    0.728     0.347      0.724     0.728      0.718      0.69

=== Confusion Matrix ===

   a   b   <-- classified as
 152  25 |   a = L
  54  59 |   b = V
```

Ilustración 18: Resultados J48 - Experimento 1

La tasa de acierto obtenida es del 72,75%, y el árbol de decisión generado por el algoritmo puede consultarse en el anexo 3.

- **Reglas - JRip:**

Para generar el conjunto de reglas, se activa el criterio de parada de ratio de error superior a 0,5, y se establecen 5 iteraciones de optimización.

```
Correctly Classified Instances         211               72.7586 %
Incorrectly Classified Instances        79               27.2414 %
Kappa statistic                          0.4054
Mean absolute error                      0.4056
Root mean squared error                  0.4451
Relative absolute error                 84.4702 %
Root relative squared error             91.1844 %
Total Number of Instances              290

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.842     0.451      0.745     0.842      0.79       0.695    L
                 0.549     0.158      0.689     0.549      0.611      0.695    V
Weighted Avg.    0.728     0.337      0.723     0.728      0.72       0.695

=== Confusion Matrix ===

   a   b   <-- classified as
 149  28 |   a = L
  51  62 |   b = V
```

Ilustración 19: Resultados JRip - Experimento 1

Un primer análisis nos permite observar que el porcentaje de acierto es el mismo que el obtenido mediante el algoritmo J48. Sin embargo, el estadístico Kappa y las medidas de error varían.

El conjunto de reglas lógicas obtenido es: *

- (Cuota Visitante ≤ 0.031211) => Gana Visitante
- (Cuota Local ≥ 0.021995) and (PPP-Recibidos-L ≥ 0.510417) and (Rebotes-Totales-L-OP <= 0.386555) => Gana Visitante
- Cualquier otro caso => Gana Local

***Valos numéricos normalizados en el intervalo [0,1]**

Tabla 7: Reglas generadas por JRip - Experimento 1

• **Bagging:** Con esta técnica, el conjunto de entrenamiento se divide en T subconjuntos con reemplazo, como se explicó anteriormente. Además para evitar empates en la votación, el número de subconjuntos sea establecido en 201, evitando que sea un número par. El algoritmo que realizará todas las votaciones será un árbol de decisión J48.

```
Correctly Classified Instances         209               72.069  %
Incorrectly Classified Instances        81               27.931  %
Kappa statistic                          0.4319
Mean absolute error                      0.3027
Root mean squared error                  0.4799
Relative absolute error                 63.0458 %
Root relative squared error             98.3278 %
Total Number of Instances              290

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.712     0.265      0.808     0.712     0.757      0.775    L
                 0.735     0.288      0.619     0.735     0.672      0.774    V
Weighted Avg.    0.721     0.274      0.734     0.721     0.724      0.775

=== Confusion Matrix ===

   a   b   <-- classified as
 126  51 |   a = L
  30  83 |   b = V
```

Ilustración 20: Resultados Bagging - Experimento 1

Con un breve análisis, se observa que esta técnica no ha mejorado el rendimiento de otros algoritmos simples como J48, siendo su tasa de acierto del 72.06%

• **Boosting:** Para un total de 51 iteraciones, la técnica no mejora los métodos individuales ya empleados. El mejor resultado ha sido obtenido aplicando la técnica sobre el clasificador Naive Bayes, obteniendo de nuevo un 72,75%.

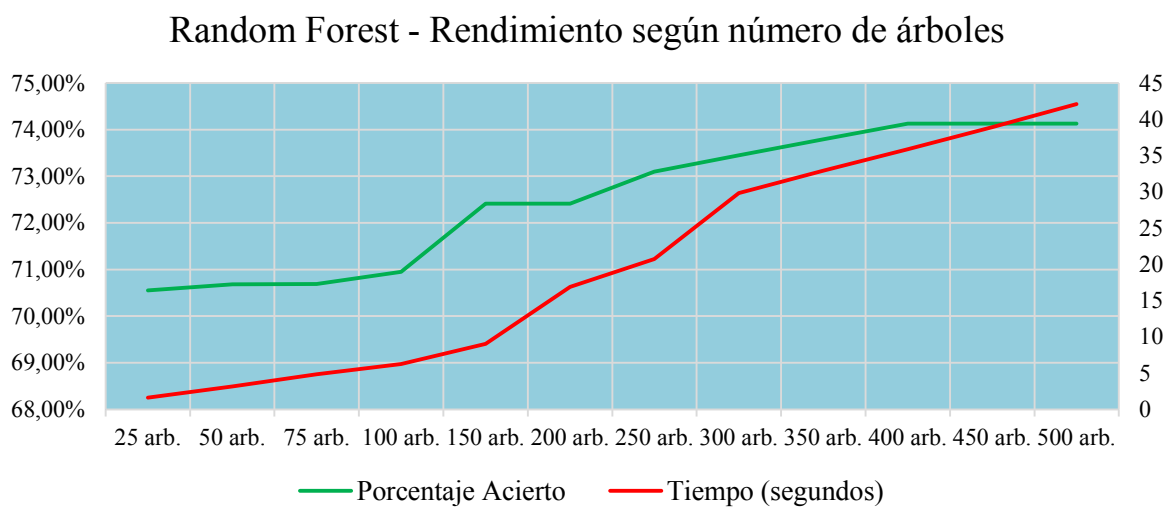
```
Correctly Classified Instances         211               72.7586 %
Incorrectly Classified Instances        79               27.2414 %
Kappa statistic                          0.4459
Mean absolute error                      0.3796
Root mean squared error                  0.4411
Relative absolute error                 79.0498 %
Root relative squared error             90.3679 %
Total Number of Instances              290

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.718    0.257      0.814     0.718     0.763      0.766    L
                 0.743    0.282      0.627     0.743     0.68       0.766    V
Weighted Avg.    0.728    0.267      0.741     0.728     0.731      0.766

=== Confusion Matrix ===

   a   b   <-- classified as
 127  50 |   a = L
  29  84 |   b = V
```

Ilustración 21: Resultados Boosting - Experimento 1

• **Random Forest:** Para este método se va a realizar en primer lugar un estudio del aumento en el porcentaje de acierto en función del número de árboles empleados. El objetivo de esto es encontrar el número de árboles a partir del cual el porcentaje de acierto se estabiliza o mejora en cantidades despreciables. Se debe recordar, que en un problema de clasificación, el número recomendable de atributos a escoger aleatoriamente es igual a la raíz cuadrada de los atributos. Por tanto, se aplicará la raíz cuadrada aproximada de 47, es decir, 7.

Random Forest		
Número de árboles	Porcentaje Acierto	Tiempo (segundos)
25 arb.	70,55%	1,63
50 arb.	70,68%	3,15
75 arb.	70,69%	4,84
100 arb.	70,95%	6,24
150 arb.	72,41%	9,03
200 arb.	72,41%	16,9
250 arb.	73,10%	20,75
300 arb.	73,45%	29,81
350 arb.	73,79%	32,9
400 arb.	74,13%	35,89
450 arb.	74,13%	38,9
500 arb.	74,13%	42,1

Tabla 8: Estudio Random Forest - Nº de árboles y Porcentaje de acierto



Gráfica 4: Random Forest - Rendimiento según Nº de árboles

Se comprueba que a partir de los 400 árboles la tasa de acierto se estabiliza. Por ello, el número de árboles escogido será 400, y los resultados obtenidos son los siguientes:

```
Correctly Classified Instances         215               74.1379 %
Incorrectly Classified Instances        75               25.8621 %
Kappa statistic                          0.4317
Mean absolute error                      0.3854
Root mean squared error                  0.4353
Relative absolute error                 80.2631 %
Root relative squared error             89.1798 %
Total Number of Instances              290

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.864     0.451      0.75      0.864     0.803      0.766    L
                 0.549     0.136      0.721     0.549     0.623      0.766    V
Weighted Avg.    0.741     0.328      0.739     0.741     0.733      0.766

=== Confusion Matrix ===

   a   b   <-- classified as
 153  24 |   a = L
  51  62 |   b = V
```

Ilustración 22: Resultados Random Forest - Experimento 1

Se observa una alta tasa de acierto del 74,13%, es decir, 215 aciertos en los 290 partidos.

• **Stacking:** Para esta técnica se van a emplear los algoritmos JRip, SMO, Naive Bayes y Random Forest (con 100 árboles), mientras que el metaclasificador será Logistic Regresion:

```
Correctly Classified Instances         213               73.4483 %
Incorrectly Classified Instances        77               26.5517 %
Kappa statistic                          0.4262
Mean absolute error                      0.4002
Root mean squared error                  0.4331
Relative absolute error                 83.3578 %
Root relative squared error             88.7294 %
Total Number of Instances              290

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.831     0.416      0.758     0.831     0.792      0.779    L
                 0.584     0.169      0.688     0.584     0.632      0.779    V
Weighted Avg.    0.734     0.32       0.73      0.734     0.73       0.779

=== Confusion Matrix ===

   a   b   <-- classified as
 147  30 |   a = L
  47  66 |   b = V
```

Ilustración 23: Resultados Stacking - Experimento 1

Para éste último método, se obtiene una tasa de acierto del 73.44%.

La recopilación y evaluación de todos los datos obtenidos mediante la aplicación de todas las técnicas empleadas en este apartado se encuentran en el apartado 5.1.

4.3 Experimentación: Primer Equipo en Anotar.

En este apartado se van a entrenar los clasificadores para el problema de predicción de qué equipo anotará primero. Se debe recordar, que en el estudio de relevancia de atributos, los resultados no fueron esperanzadores, y por tanto, se esperan unos resultados bajos.

En primer lugar se emplearán los atributos específicos del problema, correspondientes al apartado 3.4.3, los cuales son:

1	1ª-Canasta
2	100-Primera-Posesión-L
3	100-Primer-tiro-L
4	100-1º-en-Anotar-L
5	100-Primera-Posesión-V
6	100-Primer-tiro-V
7	100-1º-en-anotar-V

Tabla 9: Atributos Experimento 2

Los resultados obtenidos para los distintos clasificadores son:

	Porcentaje de Acierto	Estadístico Kappa
Naive Bayes	44,85%	-0,1164
SVM	51,49%	0,0149
JRIP	47,50%	-0,0652
J48	51,49%	0
Perceptrón Multicapa	47,17%	-0,0551

Tabla 10: Resultados Experimento 2 ; fase 1

Estos resultados tan bajos nos demuestran que con este conjunto de datos, la predicción se reduce a probabilidades cercanas a un lanzamiento de moneda. Además el estadístico Kappa nos muestra la nula correlación que tienen los atributos con la clase. Esto nos invita a realizar más pruebas, pero esta vez con un conjunto de datos más grande.

Como el conjunto de instancias para este experimento es notablemente menor, no se va a dividir el mismo en entrenamiento y test, sino que se empleará la técnica de validación cruzada con 10 iteraciones. Es decir, se divide el conjunto en 10 grupos y se utiliza cada uno como test, utilizando los restantes para entrenamiento. Después se calcula la media de los resultados obtenidos.

A continuación, se muestran los resultados aplicando todos los algoritmos descritos, pero esta vez añadiendo al conjunto de datos todos aquellos empleados en el problema de predicción de qué equipo ganará el partido. Las configuraciones de los modelos son iguales que las del apartado anterior, por tanto se muestran directamente los resultados.

- **Naïve Bayes:**

```
Correctly Classified Instances         166               55.1495 %
Incorrectly Classified Instances       135               44.8505 %
Kappa statistic                          0.0786
Mean absolute error                      0.4807
Root mean squared error                  0.4942
Relative absolute error                 96.2218 %
Root relative squared error             98.8673 %
Total Number of Instances              301

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.096     0.019      0.824     0.096      0.172      0.525    V
                 0.981     0.904      0.535     0.981      0.692      0.525    L
Weighted Avg.    0.551     0.475      0.675     0.551      0.44       0.525

=== Confusion Matrix ===

   a   b   <-- classified as
  14 132 |   a = V
   3 152 |   b = L
```

Ilustración 24: Resultados Naïve Bayes - Experimento 2

Mediante esta técnica el porcentaje de acierto aumenta al 55,14%.

- **SVM:**

```
Correctly Classified Instances         167               55.4817 %
Incorrectly Classified Instances       134               44.5183 %
Kappa statistic                          0.1035
Mean absolute error                      0.4452
Root mean squared error                  0.6672
Relative absolute error                 89.106  %
Root relative squared error            133.4885 %
Total Number of Instances              301

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.438     0.335      0.552     0.438     0.489      0.551    V
                 0.665     0.562      0.557     0.665     0.606      0.551    L
Weighted Avg.    0.555     0.452      0.554     0.555     0.549      0.551

=== Confusion Matrix ===

   a   b   <-- classified as
 64  82 |   a = V
 52 103 |   b = L
```

Ilustración 25: Resultados SVM - Experimento 2

Empleando SVM se obtiene un 55,48% de acierto.

- **JRip:**

```
Correctly Classified Instances         154               51.1628 %
Incorrectly Classified Instances       147               48.8372 %
Kappa statistic                          0.0159
Mean absolute error                      0.4968
Root mean squared error                  0.5189
Relative absolute error                 99.4464 %
Root relative squared error            103.813  %
Total Number of Instances              301

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.384     0.368      0.496     0.384     0.432      0.506    V
                 0.632     0.616      0.521     0.632     0.571      0.506    L
Weighted Avg.    0.512     0.496      0.509     0.512     0.504      0.506

=== Confusion Matrix ===

  a  b   <-- classified as
 56 90 |  a = V
 57 98 |  b = L
```

Ilustración 26: Resultados JRip - Experimento 2

En JRip la mejora solo llega al 51,16%, lo cual sigue siendo insuficiente.

- **J48:**

```
Correctly Classified Instances         156               51.8272 %
Incorrectly Classified Instances       145               48.1728 %
Kappa statistic                          0.034
Mean absolute error                      0.4888
Root mean squared error                  0.678
Relative absolute error                 97.8346 %
Root relative squared error            135.6481 %
Total Number of Instances              301

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.473     0.439      0.504     0.473     0.488      0.504    V
                 0.561     0.527      0.53      0.561     0.545      0.504    L
Weighted Avg.    0.518     0.484      0.517     0.518     0.517      0.504

=== Confusion Matrix ===

  a  b   <-- classified as
 69 77 |  a = V
 68 87 |  b = L
```

Ilustración 27: Resultados J48 - Experimento 2

Mediante J48 se obtiene un porcentaje del 51,82%, el cual sigue siendo bajo.

- **Perceptrón Multicapa:**

```
Correctly Classified Instances         171               56.8106 %
Incorrectly Classified Instances       130               43.1894 %
Kappa statistic                          0.1361
Mean absolute error                      0.4264
Root mean squared error                  0.5946
Relative absolute error                 85.342  %
Root relative squared error            118.9583 %
Total Number of Instances              301

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.568     0.432      0.553     0.568     0.561      0.602    V
                 0.568     0.432      0.583     0.568     0.575      0.602    L
Weighted Avg.    0.568     0.432      0.568     0.568     0.568      0.602

=== Confusion Matrix ===

  a  b   <-- classified as
 83 63 |  a = V
 67 88 |  b = L
```

Ilustración 28: Resultados Red de Neuronas - Experimento 2

Mediante la red de neuronas se consigue escalar a un 56,81%. A pesar de seguir siendo bajo, se comprueba que la inclusión de estos nuevos atributos es positiva.

- **Bagging:**

```
Correctly Classified Instances         162               53.8206 %
Incorrectly Classified Instances       139               46.1794 %
Kappa statistic                          0.0743
Mean absolute error                      0.4793
Root mean squared error                  0.5234
Relative absolute error                 95.936  %
Root relative squared error            104.7065 %
Total Number of Instances              301

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.5       0.426      0.525     0.5       0.512      0.547    V
                 0.574     0.5        0.549     0.574     0.562      0.547    L
Weighted Avg.    0.538     0.464      0.538     0.538     0.538      0.547

=== Confusion Matrix ===

  a  b   <-- classified as
 73 73 |  a = V
 66 89 |  b = L
```

Ilustración 29: Resultados Bagging - Experimento 2

Empleando Bagging se obtiene un 53,83% de acierto.

- **AdaBoost:**

```
Correctly Classified Instances         160               53.1561 %
Incorrectly Classified Instances       141               46.8439 %
Kappa statistic                          0.0599
Mean absolute error                      0.477
Root mean squared error                  0.6752
Relative absolute error                 95.4783 %
Root relative squared error            135.0885 %
Total Number of Instances              301

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.473     0.413      0.519     0.473     0.495      0.497    V
                 0.587     0.527      0.542     0.587     0.563      0.497    L
Weighted Avg.    0.532     0.472      0.531     0.532     0.53       0.497

=== Confusion Matrix ===

  a  b   <-- classified as
 69 77 |  a = V
 64 91 |  b = L
```

Ilustración 30: Resultados AdaBoost - Experimento 2

AdaBoost ofrece un 53,15% de partidos correctamente clasificados.

- **Random Forest:**

```
Correctly Classified Instances         158               52.4917 %
Incorrectly Classified Instances       143               47.5083 %
Kappa statistic                          0.0514
Mean absolute error                      0.492
Root mean squared error                  0.5168
Relative absolute error                 98.4821 %
Root relative squared error            103.3997 %
Total Number of Instances              301

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.555     0.503      0.509     0.555      0.531      0.523    V
                 0.497     0.445      0.542     0.497      0.519      0.523    L
Weighted Avg.    0.525     0.473      0.526     0.525      0.525      0.523

=== Confusion Matrix ===

  a  b   <-- classified as
 81 65 |  a = V
 78 77 |  b = L
```

Ilustración 31: Resultados Random Forest - Experimento 2

A pesar de que Random Forest ofreció un alto porcentaje de acierto en el experimento anterior, en este solo alcanza el 52,49%.

- **Stacking:**

```
Correctly Classified Instances         159               52.8239 %
Incorrectly Classified Instances       142               47.1761 %
Kappa statistic                          0.0553
Mean absolute error                      0.5003
Root mean squared error                  0.5784
Relative absolute error                100.1439 %
Root relative squared error            115.7239 %
Total Number of Instances              301

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.507     0.452      0.514     0.507      0.51       0.503    V
                 0.548     0.493      0.541     0.548      0.545      0.503    L
Weighted Avg.    0.528     0.473      0.528     0.528      0.528      0.503

=== Confusion Matrix ===

  a  b   <-- classified as
 74 72 |  a = V
 70 85 |  b = L
```

Ilustración 32: Resultados Stacking - Experimento 2

Por último, la técnica Stacking con la misma configuración que en el experimento anterior, solo ofrece un 52,82% de acierto.

La recopilación y evaluación de todos los datos obtenidos en este apartado para este experimento se encuentran en el apartado 5.4.

5. Resultados y evaluación de los modelos

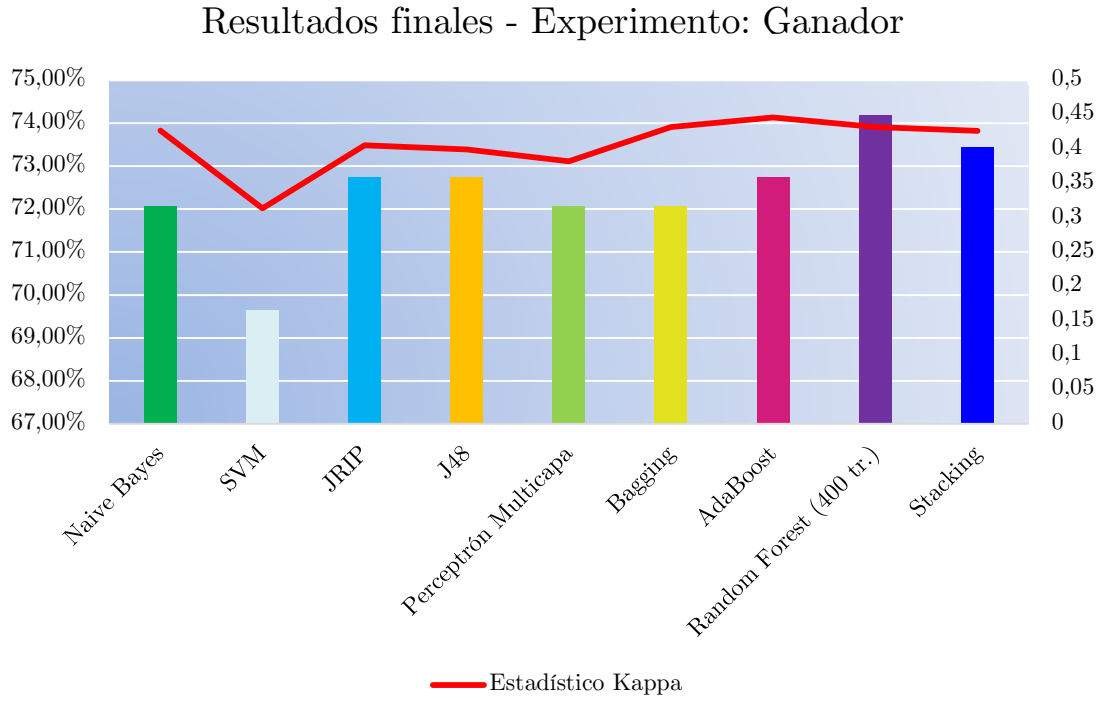
En este apartado se recopilan todos los datos obtenidos en el punto anterior, con el fin de realizar un análisis que ofrezca conclusiones relevantes. La discusión de estos resultados para el primer experimento abarca los puntos 5.1, 5.2 y 5.3. Por otro lado, el experimento de predicción del primer equipo en anotar se discute en el apartado 5.4

5.1 Evaluación experimento: Ganador

A continuación se muestra una recopilación de los datos obtenidos para este experimento.

Experimento: Ganador		
	Porcentaje Acierto	**Estadístico Kappa**
Naive Bayes	72,06%	0,4265
SVM	69,65%	0,3137
JRIP	72,75%	0,4054
J48	72,75%	0,3994
Perceptrón Multicapa	72,06%	0,3821
Bagging	72,06%	0,4319
AdaBoost	72,75%	0,4459
Random Forest (400 tr.)	74,13%	0,4317
Stacking	73,44%	0,4262

Tabla 11: Resultados Experimento 1



Gráfica 5: Resultados finales - Experimento 1

La técnica Random Forest ha sido la que ha ofrecido el porcentaje de acierto más alto alcanzando un 74,13%, lo que supone clasificar correctamente 215 partidos de 290. Además, las técnicas JRIP y J48 alcanzan un 72,75%. Por tanto, este problema muestra una buena adaptación a algoritmos basados en árboles de decisión y reglas.

Las dos técnicas que requirieron mayor tiempo de computación fueron Random Forest y Stacking, y son precisamente las dos técnicas con mayor porcentaje de acierto.

A pesar de que alcanzar un 74,13% de acierto implique clasificar correctamente algo menos de 3 partidos de cada 4, la explotación del modelo en una casa de apuesta no garantiza el éxito. Los beneficios dependerán de qué tipo de partidos son los que el clasificador está acertando, y cuales los que está

fallando. Si el modelo solo es capaz de acertar aquellos partidos con probabilidad muy alta, es probable que se pierda dinero.

5.2 Experimentación adicional y evaluación del problema del Ganador

En este apartado se realizan dos experimentos alternativos al primero. En el primero se va a comprobar cómo evoluciona la tasa de acierto durante la temporada, y en el segundo se va a estudiar el impacto de los atributos que incluyen las cuotas de las casas de apuestas.

- **Mejora en la clasificación a lo largo de la temporada**

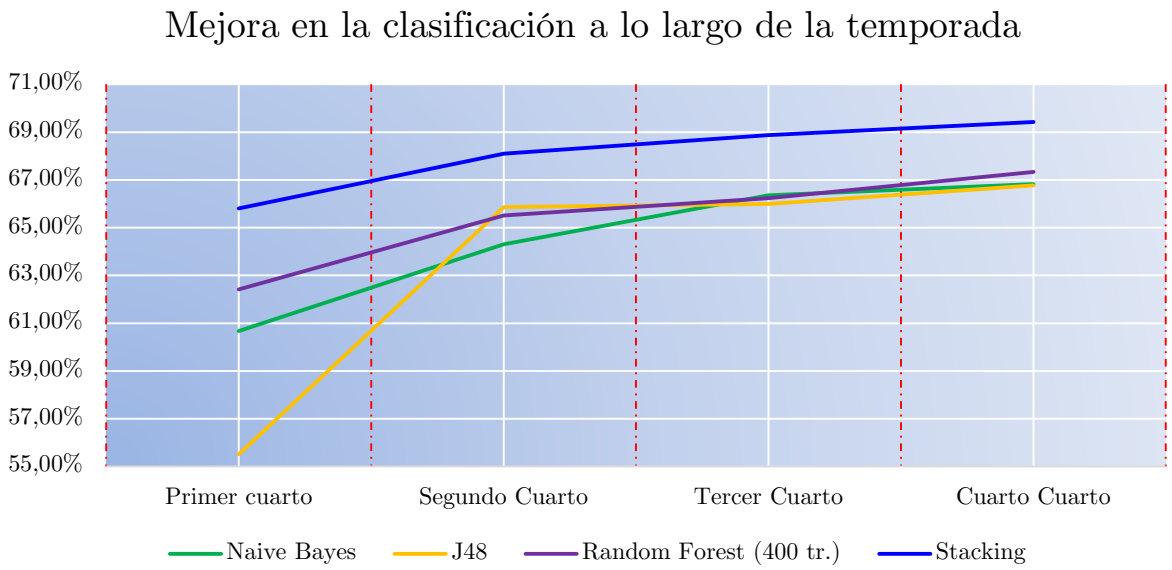
En este apartado se estudia cómo evoluciona el rendimiento de los modelos en el tiempo. Para ello se ha dividido el conjunto total de partidos en cuatro cuartos, siguiendo un orden cronológico de partidos. Los resultados deben mostrar fuertes variaciones al comienzo de la temporada, y mostrar cierta convergencia a medida que el fin de la temporada se aproxima.

Es necesario destacar que para comprobar la evolución se ha utilizado la técnica de validación cruzada. Por ello, es posible que las tasas varíen respecto al apartado anterior, y que sean otros modelos los que ofrezcan un mejor resultado que Random Forest. Sin embargo, el objetivo de este apartado consiste en estudiar la evolución de los clasificadores, y no obtener un mejor resultado final.

Para el experimento se han elegido cuatro de los algoritmos con mejores resultados en el apartado anterior, y los resultados son los siguientes:

Mejora de la clasificación durante la temporada				
	Primer cuarto	Segundo Cuarto	Tercer Cuarto	Cuarto Cuarto
Naive Bayes	60,68%	64,31%	66,36%	66,83%
J48	55,51%	65,86%	65,99%	66,77%
Random Forest (400 tr.)	62,41%	65,51%	66,24%	67,33%
Stacking	65,80%	68,10%	68,88%	69,42%

Tabla 12: Evolución del porcentaje de acierto en la temporada



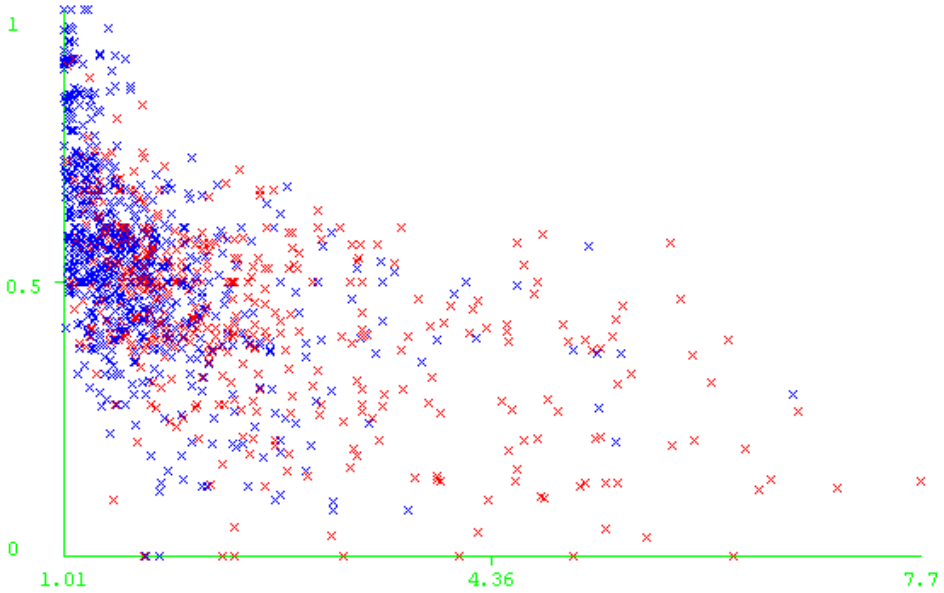
Gráfica 6: Evolución del porcentaje de acierto en la temporada

Como se puede comprobar, las predicciones han resultado ciertas, las tasas a comienzo de temporada son realmente bajas en comparación con las del final de la misma.

Otra conclusión que nos invita a reflexionar este resultado es que, si las temporadas fuesen mucho más largas, los rendimientos de los equipos se estabilizarían en su media, y por tanto, las tasas de acierto incrementarían notablemente.

• **Exclusión de atributos no deportivos.**

La siguiente gráfica muestra las 290 instancias del conjunto de test. En azul, los partidos en los que acabó ganando el equipo local, y en rojo los equipos que ganó el equipo visitante. El eje x representa la cuota de casa de apuestas para el equipo local, y el y el porcentaje de victorias del equipo local.



Gráfica 7: Distribución datos de test para atributos: Cuota L y Porcentaje Victorias L

Como se puede comprobar los partidos en los que la cuota para el equipo local era muy baja han sido mayoritariamente ganados por el equipo local. Y, además, ese equipo tenía un porcentaje de victorias en la temporada mayor del 50% en la mayoría de los casos. Sin embargo, como se explicó anteriormente, el equipo local gana aproximadamente un 60% de ocasiones, por ello resulta normal que la tendencia se extienda hasta los equipos que poseen cerca de un 40% de victorias. Además, se observa que según la cuota local aumenta los partidos suelen ser ganados por el equipo visitante.

Estos resultados nos invitan a pensar que los atributos que recogen las cuotas de las casas de apuestas para los partidos son un buen indicador global

que ya recoge el rendimiento de ambos equipos. Para comprobar la importancia de estos atributos se realiza a continuación el mismo experimento de predicción para clasificar qué equipo ganará un partido. Pero esta vez, se excluyen los atributos referentes a las cuotas, realizando el estudio con atributos estrictamente deportivos.

Estudio: Ganador con atributos estrictamente deportivos		
	Porc. Acierto Sin Cuotas	**Porc. Acierto Con cuotas**
Naive Bayes	68,62%	72,06%
SVM	68,27%	69,65%
JRIP	66,20%	72,75%
J48	67,58%	72,75%
Perceptrón Multicapa	69,31%	72,06%
Bagging	66,89%	72,06%
AdaBoost	65,44%	72,75%
Random Forest (400 tr.)	69,54%	74,19%
Stacking	67,93%	73,44%

Tabla 13: Porcentaje de acierto excluyendo cuotas de apuestas

Como puede comprobarse con el conjunto de reglas obtenido mediante JRip*, ahora los atributos que forman las reglas son estrictamente deportivos:

(Porc. VictoriasL ≤ 0.425) and (Porc. VictoriasV ≥ 0.462) -> Gana Visitante

(Tapones V OP ≤ 0.471698) and (Porc. Victorias L ≤ 0.595) and (Asistencias V ≥ 0.458599) -> Gana Visitante

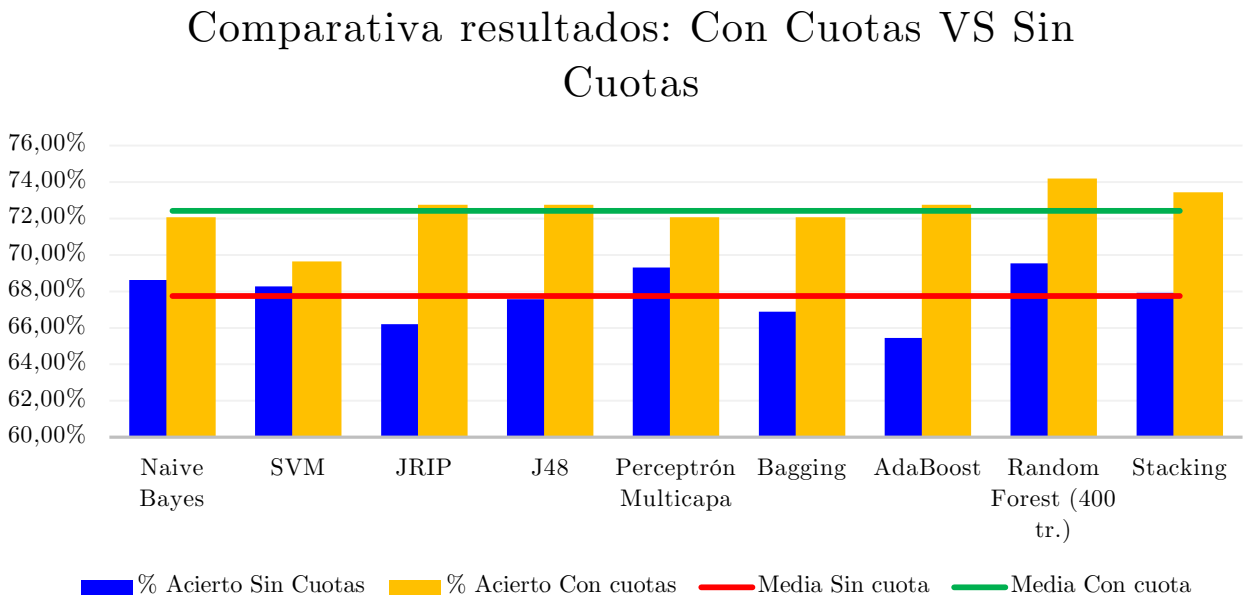
(Rebotes-Totales-V ≥ 0.436364) and (Tiros-Anotados-L ≤ 0.428571) and (Tiros-Anotados-L-OP ≥ 0.659574) => Gana Visitante

Cualquier otro caso => Gana Local

***Valos numéricos normalizados en el intervalo [0,1]**

Tabla 14: Reglas generadas por JRip - Cuotas de apuestas excluidas

A continuación se muestra una gráfica comparativa de ambos experimentos y se remarcan las medias obtenidas en las tasas de acierto.



Gráfica 8: Comparativa resultados Con cuotas vs Sin cuotas

La conclusión obtenida con este estudio, es que los atributos que reflejan la cuota de las casas de apuestas influyen notoriamente en la predicción. La media de acierto ha disminuido de un 72,41% utilizando todos los atributos a un 67,75% si solo se utilizan atributos estrictamente deportivos.

5.3 Comparación de resultados del problema del Ganador con otros proyectos similares

Como se explicó en el estudio de proyectos similares, destacan cuatro estudios realizados sobre la misma materia. En este apartado se comparan los resultados obtenidos en este proyecto con estos estudios, procurando realizar comparaciones justas y adaptadas a los diferentes dominios.

El dato conocido para Accuscore corresponde a la temporada 2013, y refleja una tasa de acierto del 70,3%. Por otra parte, la mejor tasa de acierto obtenida en este proyecto ha sido 74,19%. A pesar de que las tasas difieren en un 3,89%, no es justo realizar una comparativa real sobre dominios que no son exactamente iguales. Es probable que gran parte de los atributos utilizados en uno y otro proyecto sean los mismos, pero no sus valores. Las diferencias entre la temporada 2013 y la temporada 2015 pueden haber sido notables, y existen diversos factores que pueden provocarlas, como una mayor igualdad entre los equipos o una notoria influencia de resultados atípicos.

Respecto al proyecto Predicting NBA Winners Project, la diferencia entre su tasa del 65,2% y la del 74,19% es notoria. Para entender esta diferencia es necesario recordar la lista de atributos empleados en el proyecto. Mientras que en este proyecto se empleaban 47 atributos tras el estudio de relevancia, en el suyo solo se utilizaban 17. Sin embargo, es más justo comparar su tasa de acierto con la tasa de acierto obtenida sin tener en cuenta las cuotas de las casas de apuestas. De este modo, en ambos proyectos se estaría intentando encontrar la solución al problema con la misma restricción: emplear únicamente atributos estrictamente deportivos. En este proyecto esa tasa es del 69,54%, lo que supone un 4,3% más. En cualquier caso, es importante recordar que los dominios no son iguales, y que el estudio fue realizado a lo largo de otra temporada, lo cual puede provocar notables diferencias.

Por otro lado, el proyecto de Cao C. refleja un 69,67% empleando atributos que recojan el rendimiento de los equipos en los 10 partidos previos al objetivo. Esta tasa supera ligeramente el 69,54% obtenido en este proyecto cuando no se emplea las cuotas de las casas de apuestas como atributo. Esta comparativa demuestra que, si un clasificador va a basarse en datos deportivos, es más relevante que recoja datos de la misma temporada en la que se está

realizando la predicción, ya que la información que puedan aportar rendimientos de temporadas anteriores no es alto.

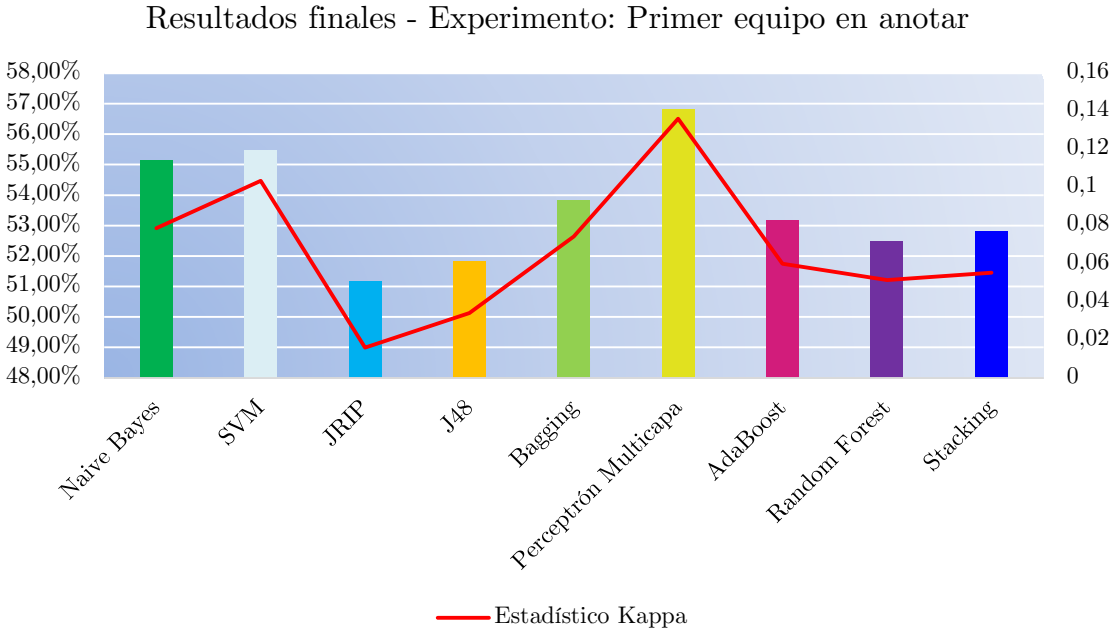
Por último, el proyecto de Loeffelholz, B. alcanzó el 74,33%, el cual si compite con el 74,19% alcanzado en este proyecto. Sin embargo, se debe recordar que el conjunto de datos del proyecto de Loeffelholz consiste únicamente en los primeros 650 partidos de la temporada 2007, empleando un pequeño conjunto de test de 30 partidos. La comparación podría ser más justa si el conjunto de datos abarcase toda la temporada, y más aún si los dominios de ambos proyectos se situaran en el mismo año. Además, existen subconjuntos de 30 partidos del conjunto de entrenamiento de nuestro proyecto que alcanzan tasas más altas. Por ejemplo, entre los 30 partidos jugados ente el 3 de abril de 2016 y el 8 de abril de 2016, el modelo desarrollado en este proyecto alcanza el 83,33% de acierto, como puede comprobarse en el anexo 4 (El cual ofrece información relevante para el apartado 6). Este hecho provoca dudas sobre la capacidad de clasificación real del proyecto de Loeffelholz.

5.4 Evaluación experimento: Primero equipo en anotar

La recopilación de los resultados obtenidos para el problema de clasificación del primer en equipo en anotar son los siguientes:

	% Acierto	**Estadístico Kappa**
Naive Bayes	55,14%	0,0786
SVM	55,48%	0,1035
JRIP	51,16%	0,0159
J48	51,82%	0,034
Perceptrón Multicapa	56,81%	0,1361
Bagging	53.82%	0.0743
AdaBoost	53,15%	0,0599
Random Forest	52,49%	0,0514
Stacking	52,82%	0,0553

Tabla 15: Resultados finales Experimento 2



Gráfica 9: Resultados finales: Experimento 2

El mejor resultado se ha obtenido mediante la aplicación del perceptrón multicapa. Sin embargo, 56,8% sigue siendo un resultado pobre. Del cual no es posible realizar una explotación económica apostando.

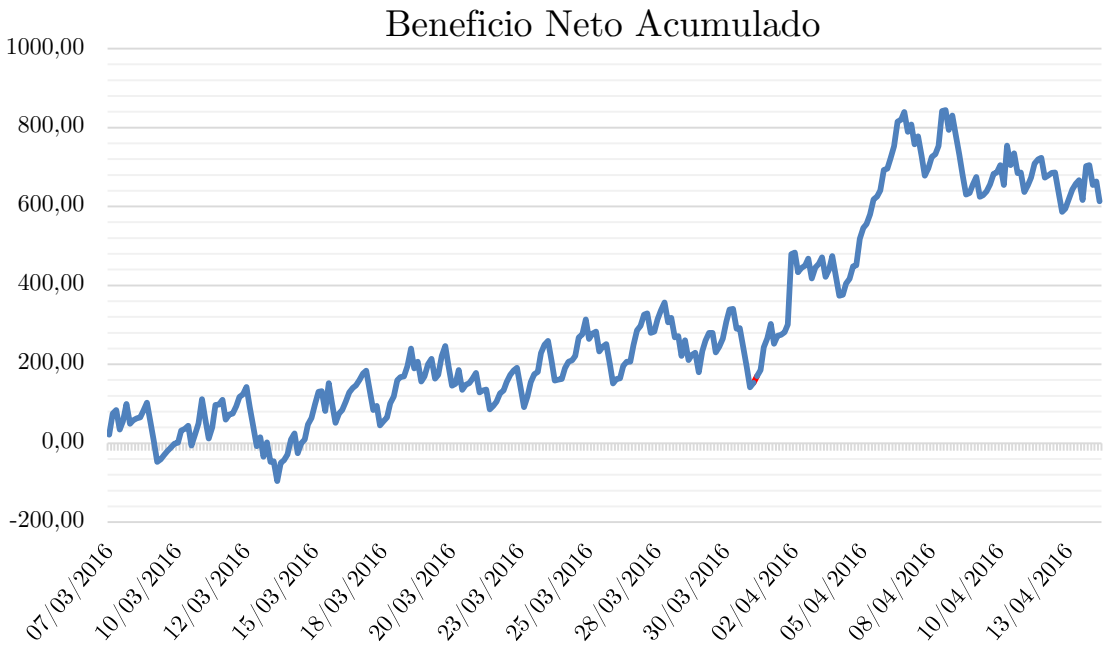
Por otra parte, cabe destacar que este estudio solo se realizó con 302 instancias debido a la complejidad de extracción de los datos. Esto provoca la apertura de una línea futura, que invita a realizar el estudio durante toda una temporada. Sin embargo, como se comprobó en los estudios de relevancia de atributos, ninguno superaba la puntuación de 0, por tanto resulta poco probable que estas tasas de acierto puedan aumentar considerablemente.

6. Explotación del modelo

El objetivo de este apartado es realizar una simulación de apuestas en base a nuestro mejor modelo (Random Forest, 400 árboles, 74,13% acierto). Con él sabemos que en el último cuarto de la temporada se predijeron correctamente 215 partidos e incorrectamente 75.

En el anexo 4, se incluye una tabla que incluye todo el proceso de explotación. Cada fila representa un partido, incluyendo las cuotas ofrecidas por las casas de apuestas, la predicción del modelo, el resultado real, el beneficio neto obtenido apostando 10 euros a ese partido, la probabilidad de acierto ofrecida por el modelo y el beneficio acumulado hasta esa fecha. Como puede comprobarse al concluir el partido número 290, el beneficio neto acumulado es de 122,7 € (Apostando 10€ por partido a todo aquello que prediga el modelo).

A continuación se incluye una tabla que muestra el progreso del beneficio neto acumulado desde el 7 de marzo de 2016 al 13 de abril de 2016:



Gráfica 10: Evolución del beneficio neto acumulado

Sin embargo, al aumentar el dinero apostado por partido, aumenta el beneficio neto total. Por ejemplo, si se apuestan 50 € a cada predicción, el beneficio neto total aumenta a 613,5 €. Por tanto, resulta interesante estudiar cómo se han distribuido las apuestas.

Analizando la tabla del anexo 4, se observa que la cuota media de los partidos clasificados correctamente es de 1,40. Por otro lado, la cuota media de los partidos clasificados incorrectamente es de 1,59. Por otro lado, sabemos que el ratio fallos/aciertos es igual a 75/215 = 0,348. Por tanto:

$$Caja\ Final = (215\ x\ 1{,}348) - 75 = 214{,}82\ €$$

$$Beneficio\ Neto\ Total = 214{,}82 - 215 = -\ 0{,}82\ €$$

Es decir, si la cuota media de los partidos clasificados correctamente fuese 1,348, el beneficio neto total seria aproximadamente 0. Si esa cuota media de partidos clasificados correctamente fuese inferior a ese valor, se obtendrían pérdidas en la explotación.

En conclusión, la cuota media de 1,40 del modelo es superior a la cuota media que iguala las ganancias, 1,348. Este es el motivo por el que el modelo generado produce beneficios para el último cuarto de la temporada 2015/2016.

7. Planificación y Presupuesto

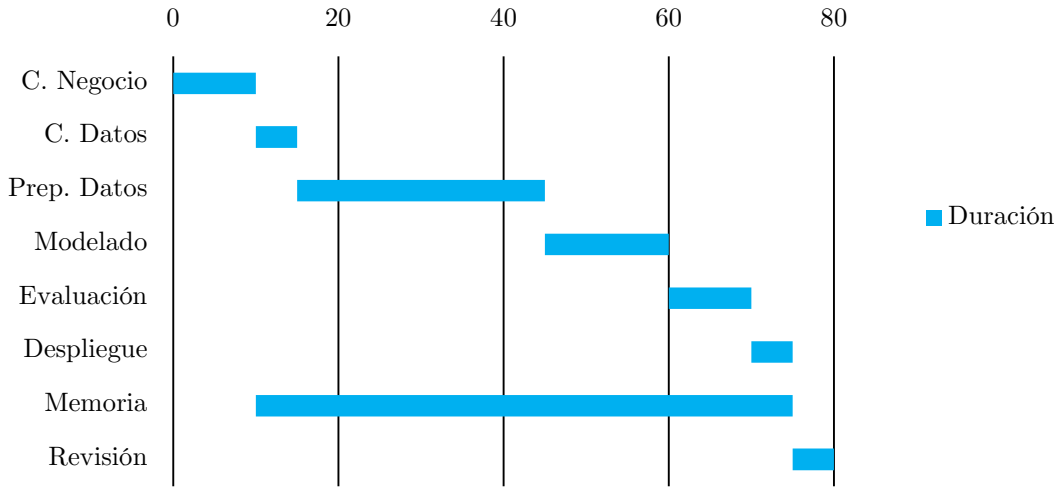
En este apartado se muestra el tiempo empleado para la realización del proyecto, y su coste. En primer lugar se detalla la planificación inicial, para posteriormente mostrar el tiempo final realmente necesitado. Las fases de la planificación se basan en la metodología implantada para este proyecto (CRISP-DM), pero además se incluyen ciertas etapas relacionadas con la confección del proyecto.

- **Planificación inicial**

    - **Comprensión del negocio:** Esta fase da comienzo al proyecto, y supone la comprensión de los objetivos, el estudio del dominio y la elección de las herramientas que darán soporte. Incluye además el estudio de los proyectos similares. Duración: 10 días.
    - **Comprensión de datos:** Consulta de repositorios de datos y definición de los métodos de extracción y adaptación de los mismos a las herramientas. Duración: 5 días.
    - **Preparación de datos:** Comprende la selección de los atributos, la extracción de los datos, los estudios de relevancia y el pre-procesado. Duración: 30 días.
    - **Modelado:** Aplicación de las técnicas de modelado. Duración: 15 días.
    - **Evaluación:** Recopilación de los datos obtenidos durante el modelado y comparación: 10 días
    - **Despliegue:** Explotación del modelo para comprobar el impacto económico fruto de su uso: 5 días.
    - **Redacción de la memoria:** Esta etapa avanza paralelamente junto al proyecto, alimentándose de los avances logrados en las otras fases. Duración: 65 días.

- **Revisión:** Se establecen 5 días de revisión para perfeccionar el proyecto.
- **Tiempo de holgura:** Tiempo reservado para ser empleado si alguna de las otras etapas sufre un retraso. Duración: 10 días.

Cada día implica una jornada de trabajo de 8 horas. Es decir, en total se han presupuestado 640 horas (excluyendo tiempo de holgura), reflejadas en el diagrama de Gantt a continuación:



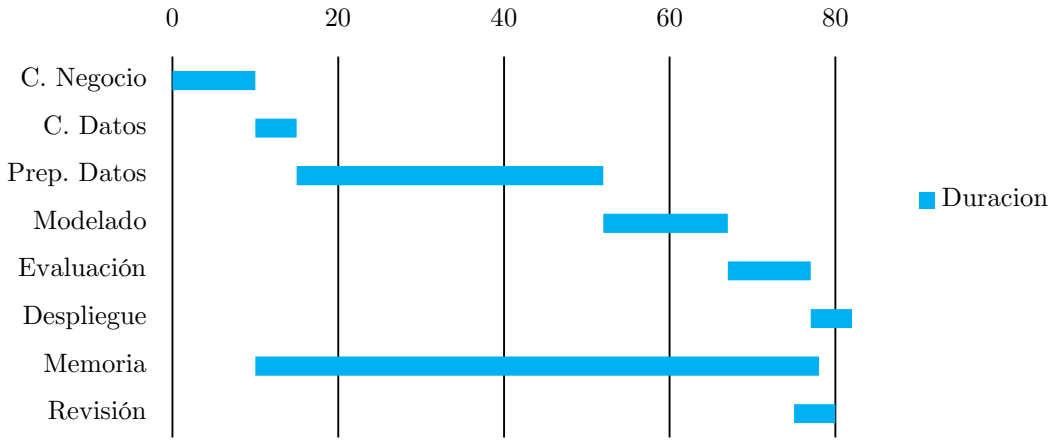
Gráfica 11: Planificación presupuestada

• Planificación final

La mayor parte de las etapas han seguido el plan establecido inicialmente. Sin embargo, para algunas ha sido necesario recurrir a los 10 días de holgura para lograr los objetivos. Estos 10 días han sido empleados en la etapa de preparación de datos. Más concretamente, en la fase de extracción de los mismos. Como se ha explicado en el proyecto, su complejidad de obtención ha provocado que gran parte del proceso se haya desarrollado manualmente. Por

otro lado el tiempo establecido para la memoria se ha alargado 3 días, necesitándose 68 días en total para completar la misma.

A continuación se muestra el diagrama de Gantt correspondiente a la planificación real:



Gráfica 12: Planificación real

• Presupuesto

A continuación se muestra el presupuesto estimado para las 640 horas planificadas inicialmente.

Personal				
Nombre	**Puesto**	**Tarifa por hora**	**Horas**	**Coste total**
Jorge Morate Vázquez	Investigador principal	27 €	640	17.280 €
			TOTAL	**17.280 €**

Tabla 16: Presupuesto personal

Equipamiento				
Descripción	**Duración**	**Amortización**	**Coste**	**Coste imputable**
Ordenador portátil Asus A53E	3,66 meses	48 meses	637 €	48,57 €
			TOTAL	**128,39 €**

Tabla 17: Presupuesto equipamiento

Software				
Descripción	**Duración**	**Amortización**	**Coste**	**Coste imputable**
Licencia Microsoft Office 2013	3,66 meses	36 meses	1.200 €	122 €
Licencia Windows 10 Pro	3,66 meses	36 meses	279 €	28,36 €
Dropbox Business 1TB	3,66 meses	12 meses	9,99 €	3,04 €
			TOTAL	**153,40 €**

Tabla 18: Presupuesto software

Resumen	Concepto	Personal	Equipamiento	Software	Total sin IVA	IVA (21%)	TOTAL
	Coste	17.280 €	128,39 €	153,40 €	**17.561,79 €**	3.687,97 €	**21.249,76 €**

Tabla 19: Presupuesto resumido

8. Marco Regulador

Como se explicó con anterioridad, los datos han sido obtenidos principalmente de dos fuentes fiables. En el repositorio Basketball Reference se pone a disposición del usuario un listado de condiciones y términos del servicio. En él se especifica que todos los datos son propiedad de la NBA, pero su consulta, distribución y uso son libres para realizar recolecciones de carácter personal, y su uso no se ve limitado en el ámbito académico.

Respecto al repositorio OddsPortal ocurre la misma circunstancia. Las consultas de las cuotas de una casa de apuestas son de libre acceso, y su recolección histórica no se ve afectada por ningún marco regulador.

9. Conclusiones y trabajo futuro

9.1 Conclusiones

Mediante el uso de técnicas de aprendizaje automático se ha dado respuesta a dos incógnitas, y se han extraído interesantes conclusiones paralelas a las mismas. En primer lugar se enumeraran las conclusiones extraídas del experimento de predicción de equipo ganador, y posteriormente las conclusiones referentes al problema de predicción del primer equipo anotador.

Tras recopilar, y pre-procesar los 1230 partidos correspondientes a la temporada NBA 2015/2016, se ha formado un conjunto de datos formado por 1161 instancias de las cuales 871 han sido empleadas para entrenar los modelos y 290 para realizar test sobre ellos.

De entre todas las técnicas empleadas Random Forest es la que ha cosechado un mejor porcentaje de acierto, llegando al 74,13%. Es decir, de los 290 partidos de test, 215 fueron clasificados correctamente, y 75 incorrectamente. La configuración óptima encontrada para el modelo es de 7 atributos aleatorios (es decir, la raíz cuadrada aproximada del total de atributos del dataset) y 400 árboles. Para cantidades de árboles superiores a 400 las mejoras en los resultados son cada vez más despreciables.

Este porcentaje de acierto, si es comparado con otros problemas de aprendizaje automático, no es alto. Predecir correctamente 3 partidos de cada 4 supone un alto margen de mejora en la materia. Sin embargo, la cantidad de situaciones, comportamientos y rendimientos aleatorios o impredecibles que rodean al dominio, implican que alcanzar tasas de acierto cercanas al 95% se antoje complicado.

Por otro lado, se ha comprobado que la mejora del porcentaje de acierto aumenta a lo largo de la temporada de manera notoria. Los resultados obtenidos

durante el primer y segundo cuarto de la temporada resultan pobres en comparación con las tasas alcanzadas al término de la temporada. Esto hace concluir que si las temporadas tuvieran un duración notablemente más larga, los rendimientos de los equipos se estabilizarían, haciendo que las predicciones fuesen más sencillas para los modelos, y por tanto, alcanzando tasas de acierto más altas. La afirmación de que la tasa de acierto aumenta cuantos más partidos se jueguen no se aplica de una temporada a otra, ya que los rendimientos de los equipos en una determinada temporada afectan muy levemente a sus rendimientos para la siguiente temporada.

A la hora de excluir aquellos atributos que no son estrictamente deportivos, se han disminuido notablemente los porcentajes de acierto. La mejor técnica en este caso ha vuelto a ser Random Forest, alcanzando un porcentaje de acierto del 69,54%. Este resultado dota a los atributos referentes a las cuotas de las casas de apuestas de una importancia mayúscula.

Las comparaciones con otros proyectos y estudios no resultan del todo justas. A pesar de que la tasa de acierto obtenida en este proyecto es la más alta, los otros proyectos trabajan sobre distintas temporadas, y en ellos no incluyen las cuotas de las casas de apuestas. Por tanto, una de las conclusiones más importantes de este proyecto es la siguiente: Si solo se toman atributos estrictamente deportivos, y referentes a las medias de los rendimientos de los equipos a lo largo del tiempo, los mejores resultados se obtienen si solo se tienen en cuenta sus actuaciones más recientes antes de la fecha de predicción. Y además, estas predicciones no superan el 69% de acierto. Es por ello que si las predicciones quieren ser realizadas únicamente mediante atributos deportivos, se debe dar un paso más e incluir atributos de obtención más compleja, como las posiciones frecuentes de tiro de los jugadores o los rendimientos ofensivos particulares de los jugadores en función de los defensores contra los que juegan.

Este proyecto incluye la explotación del mejor modelo en una casa de apuestas. El ratio fallos/acierto del mejor modelo es de 75/215=0,348. Esto implica que si la media de las cuotas de los 215 partidos es igual a 1,348, el beneficio neto generado por el modelo sería de 0 €. Sin embargo, la cuota media de los aciertos es de 1,40. Esto implica que el modelo generado cosecha un beneficio neto positivo para el último cuarto de la temporada 2015/2016.

Por último, respecto al experimento de predicción del primer equipo anotador, los resultados han sido pobres. Un perceptrón multicapa con una capa oculta y 20 neuronas ha obtenido un 56,81% de acierto realizando validación cruzada sobre 302 instancias. Ese 6,81% que difiere a la tasa de acierto con el 50% que implica aleatoriedad, invita a considerar que un mayor número de instancias pueda incrementar la tasa de acierto hasta cierto punto. Este pequeño porcentaje que supera el 50% se ha obtenido incluyendo al conjunto de datos aquellos atributos empleados en el experimento de predicción del equipo ganador. Por tanto, a pesar de que las probabilidades de que un equipo u otro anoten la primera canasta son muy parejas, la balanza se inclina levemente hacia el equipo favorito para ganar el partido.

9.2 Trabajos futuros

En primer lugar se van a estudiar los trabajos futuros referentes al experimento de predicción de equipa ganador y posteriormente al experimento de predicción de primer equipo anotador.

Como se ha explicado en el apartado anterior, el primer experimento dispone de un amplio margen de mejora, pero se ve envuelto por un dominio repleto de atributos y comportamientos humanos impredecibles. En este proyecto se ha asumido que las medidas de los rendimientos medios de los equipos son un estimador suficiente para realizar las comparaciones entre equipos. Sin embargo un análisis profundo de los rendimientos individuales puede aportar nuevas características al modelo. Los jugadores, ofensivamente hablando, tienden a realizar sus lanzamientos en ciertas zonas del campo. Por otro lado, los equipos muestras mejores capacidades defensivas en zonas del campo concretas. Por tanto, la efectividad de tiro de los equipos puede predecirse de manera más precisa si se estudian las zonas y porcentajes de tiro individualmente, y los rendimientos defensivos individuales de los rivales en esas zonas. En definitiva, es interesante comprobar cómo afectaría a la precisión del modelo la inclusión de más atributos que no se recojan en las hojas de estadísticas de los partidos.

Por otro lado, el problema de predicción de primer equipo anotador presenta ciertos márgenes de mejora. En primer lugar, es interesante crear un método automático que pueda recoger la información extraída de las hojas de jugadas de partidos, sin tener que realizarlo manualmente, lo cual es una tarea que lleva una cantidad elevada de tiempo. Esta automatización del proceso permitiría conseguir un mayor número de instancias, lo que podría suponer un

pequeño incremento en la precisión de acierto. Además, la inclusión de atributos referentes al rendimiento de los jugadores titulares (los que presumiblemente anotarán la primera canasta), resulta interesante. Por tanto, estudiar los porcentajes y las zonas de tiro individuales cuando los jugadores son defendidos por otros rivales concretos, puede incrementar levemente el éxito del modelo.

Por último, estos estudios pueden extrapolarse a otras apuestas que se encuentran presentes en el mundo del baloncesto. Algunas apuestas sobre las que aplicar experimentos de aprendizaje automático pueden ser: Ganador del partido al descanso, cantidad de puntos anotada por un jugador en concreto, o cantidad total de puntos anotada por ambos equipos.

Referencias

[1] Definición.de. (2016). Definición de baloncesto — Definicion.de. [online] Recuperado de: http://definicion.de/baloncesto/

[2] Kelbet. (2014) ¿Qué son las cuotas y cómo funcionan? - Kelbet. [online] Recuperado de: http://kelbet.es/las-cuotas-y-como-funcionan.html

[3] Brown, W. & Sauer, R. (1993). Fundamentals or noise? Evidence from the professional basketball betting market. *Journal of Finance*, 48, 1193–1209.

[4] Humphreys, B. (2010). Point spread shading and behavioral biases in NBA betting market. *Rivista Di Diritto Economia Dello Sport*, 13-26.

[5] Molero, G y Meda, M (2010). Integración de Minería de Datos y Sistemas Multiagente: Un campo de investigación y desarrollo. *Ciencias de la Información*, 41, 53-56

[6] Menendez-Barzanallana, R. (2015). *Informática Aplicada a las Ciencias Sociales. Bases de datos. Rafael Barzanallana. UMU.* [online] Um.es. Recuperado de: http://www.um.es/docencia/barzana/IACCSS/Bases-de-datos.html

[7] Politécnica de Tlaxcala (2015). Métodos predictivos y Descriptivos – Minería de Datos. (Politécnica de Tlaxcala, San Pedro Xalcaltinco, México).

[8] Schapire, R. (2008). Theoretical Machine Learning. (Computer Science lecture, Princeton University, New Jersey, United States)

[9] Caudill, M. (1989). Neural Network Primer: Part I. San Diego: Adaptics INC.

[10] Rokach, L. & Maimon, O. (2008). *Data mining with decision trees: theory and applications*. World Scientific Pub Co Inc.

[11] López, B. (2005). *Algoritmo C4.5* (Instituto Tecnológico de Laredo, Nuevo Laredo, México).

[12] Kotsiantis, S. (2007) *Supervised Machine Learning: A Review of Classification Techniques,* 249-268.

[13] Prieto, O. and Casillas, R. (s. f.). *Aprendizaje Bayesiano* [online] Recuperado de: http://www.infor.uva.es/~isaac/doctorado/AprendizajeBayesiano.pdf

[14] Mitchell, T. (1997) *Machine Learning, 6*. Pittsbirgh, PA, Estados Unidos: McGraw-Hill

[15] Malagón, C. (2003) *Clasificadores Bayesianos* (Universidad Nebrija, Madrid).

[16] Cortes, C. & Vapnik, V. (1995) Support-vector networks. *Machine Learning* 20, 273.

[17] Moujahid, A. (s. f.) *Inducción de Reglas* (Universidad del País Vasco, Donostia, Gipuzkoa).

[18] Weka Sourceforge (2016). *JRip*. [online] Recuperado de: http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/JRip.html

[19] Cohen, W. (1995) Fast effective rule induction, *Proceedings of the Twelfth International Conference On Machine Learning.* (Tahoe City, CA, USA).

[20] Dietterich, T. (2000) *Ensemble Methods in Machine Learning* (Publication, Oregon State University, Corvallis, Oregon, USA).

[21] Borrajo, M. (2014) Aprendizaje Automático. *Conjuntos de Clasificadores* (Uc3m, Madrid, España).

[22] Cárdenas-Montes, M. (2016). Bagging (CIEMAT, Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, Madrid, España).

[23] Apuntes-R (2014). Data Mining con R., Bagging para mejorar un modelo predictivo [online] Recuperado de: http://apuntes-r.blogspot.com.es/2014/12/bagging-para-mejorar-un-modelo.html

[24] Apuntes-R (2014). Data Mining con R, Random Forest. [online] Recuperado de: http://apuntes-r.blogspot.com.es/2014/11/ejemplo-de-random-forest.html

[25] Scikit-learn (s. f.) *Ensemble Methods* [online] Recuperado de: http://scikit-learn.org/stable/modules/ensemble.html

[25] Wolpert, D. H. (1992) Stacked generalization. *Neural Networks* 5. 241-259.

[26] University Of California, Los Angeles (s.f.) What is Data Mining? [online] Recuperado de: http://www.anderson.ucla.edu/faculty/jasonfrand/palace/datamining.htm

[27] Lin, J., Short, L & Sundaresan, V. (2014) *Predicting National Basketball Association Winners.* (Standford University, Stanford, California, USA).

[28] Cao, C. (2012) *Sports Data Mining Technology Used in Basketball Outcome Prediction*. (Masters Dissertation. Dublin Institute of Technology, Dublin, Ireland)

[29] Loeffelholz, B. (2009) *Predicting NBA Games Using Neural Networks*, 13. (Berkeley Electronic Press).

[30] Marbán, O. y Mariscal, G. (2009) Minería de datos y descubrimiento de conocimiento en aplicaciones reales. *Process.* 438-453 (I-Tech, Viena, Austria).

[31] University of Waikato (s. f.) *Data Mining Software in Java.* [online] Recuperado de: http://www.cs.waikato.ac.nz/ml/weka/

[32] University of Waikato (s. f.) *Attribute-Relation File Format (ARFF).* [online] Recuperado de: http://www.cs.waikato.ac.nz/ml/weka/arff.html

[33] OddsPortal (s. f.) *NBA Results & Historical Odds.* [online] Recuperado de: http://www.oddsportal.com/basketball/usa/nba/results/-/

[34] Basketball-Reference (s.f.) [online] Recuperado de: http://www.basketball-reference.com/

[35] Vryniotis, V. (2014) Machine Learning & Software Development News, *Using Feature Selection Methods in Text Classfication*

[36] Cambridge University Press. (2008) *Assessing Chi as a feature selection method.* (Cambridge, UK).

[37] Raschka, S. (2014) *About Feature Scaling and Normalization* [online] Recuperado de:

http://sebastianraschka.com/Articles/2014_about_feature_scaling.html

Anexo 1: Resumen en inglés

Abstract

The Main Goal of this project is to predict and correctly classify which team will win a specific NBA game, based on objective and previous data of the regular season 2015/2016.

Simultaneously, another experiment will be developed. The goal of this parallel project is to predict which team will score first in a particular game. This observation will prove if this fact is truly unpredictable.

To develop this experiments, different machine learning techniques will be used in order to assess the results from different approaches.

The best model obtained will be utilized betting money over a set of games. The chosen bet lines will be the ones that the generated model predicts.

Keywords: Machine Learning, WEKA, NBA, features, Data Mining, Sport Betting.

Introduction, goals and motivation.

Throughout the history of sport, and especially since XX century, results and statistics have been carefully collected for different purposes. Thanks to this collection, there is objective evidence of the performance of teams and players over time. In addition, it is possible to set records and milestones of different nature. The player's performance in this particular sport is easy to obtain numerically. This allows us to compare the performance of different teams from all the different points of view that such a large number of measurable attributes offer.

On the other hand, information science, among its many applications, has developed ways to predict events based on previous data collection. These predictions are used in numerous fields such as medicine, meteorology or biology, and some of them yield remarkably positive results. Machine learning techniques are used to achieve them, which are classified as supervised and unsupervised.

The predictions developed in this project may be useful for a basketball bettor, in order to yield economic benefit. Nevertheless, this project will not focus solely on predicting which team will win a game. In the field of basketball betting, it is possible to bet on which team will be the first to score. The bookmakers offer the same odds for both events, so it is assumed that, a priori, there is no evidence to predict that a team is more likely to score first. Therefore, this project will develop a study of the regular season 2015/2016 to try to find attributes that can help predict which team will be the first to score in a particular game.

State Of Art

• Data Mining

Data mining is a process that finds useful information that is collected on other data sets. Thanks to this process, it is possible to discover patterns in large stores of information beyond simple analysis.

Technologically, data mining has solved two major challenges: To extract interesting information from large data sets, and use techniques to explore, analyze, understand and identify patterns that help us to understand the environment and to take decisions.

Moreover, data mining has unified numerous fields in its application and during its process. The process is applicable, among other disciplines, to finance, market analysis, medicine, telecommunications, security, environmental analysis and chemistry. But during the process, different fields such as databases, statistics, artificial neural networks, machine learning and pattern recognition are interrelated in order to achieve the goals.

The steps of data mining process can be listed as follows:

- **Selection of the data set where the information is going to be obtained:** This step consists in choosing a reliable source of information that is large enough to solve the problem.

- **Analysis of the data properties:** The features that are obtained have to be analyzed in order to predict bias.

- **Transformation of the inputs:** The dataset is prepared in order to apply the techniques that best suit the problem. This requires the implementation of a step known as pre-processing.

- **Selection and application of the data mining technique chosen:** Step where the techniques are applied and the results are collected before being interpreted.

- **Interpretation and data evaluation:** First of all it is important to make sure that the results are correct in order to check if they solve the original problem. If several techniques have been applied, the results have to be compared. If at this point the results are unsatisfactory, the investigation must go back to a previous point, either to change features, to modify the data transformation, or to choose new techniques.

• Machine Learning

Machine Learning is the study of algorithms and programs that can improve automatically, basing themselves on previous experience. Its goals are to learn new knowledge and improve behaviors.

In other words, machine learning consists in improving the future using past experiences.

Moreover, the word "automatic" should be emphasized. The goal is to design learning algorithms that learn automatically without human intervention or assistance. Instead of programming the computer to solve problems, machine

learning searches for methods that allow the computer to create its own programs based on examples with which we provide it.

It is unlikely we will be able to create any kind of intelligent system that possesses any features associated with intelligence (such as speech or vision) without using learning to obtain these features.

The machine learning techniques that will be used in this project are the following:

- Artificial Neural network:
- Decision trees (J48)
- Bayesian network (Naïve Bayes)
- SVM
- Association Rules. (JRip).

Among the ensemble learning techniques, the ones that will be used are:

- Bagging
- Boosting
- Random Forest
- Stacking

Projects related to prediction applied to basketball

The NBA itself has already applied data mining to its own competition. One of the most popular applications is "Advanced Scout". This application, based on data mining, is used by the team coaches to discover interesting patterns from the box scores and recordings of the previous games. An example of this is found on January 6, 1995. New York played against Cleveland Cavaliers, and the analysis of the play-by-play statistics revealed that when Mark Price was playing as a point guard, John Williams tried four Three-Point shots, scoring all

of them. It also highlights the significant difference between John's accuracy and the team's accuracy. (49.30%)

The following projects aim to predict the outcomes of NBA Games, and they will serve as a model for trying not to make the same mistakes, and for adding new features that can improve accuracy:

- "Predicting NBA Winners" is a project produced by Jasper Lin, Logan Short and Vishu Sundaresan that uses NBA data from 1991 to 1998, and it develops a machine learning model to predict the winner in NBA games. It only includes 17 features based on the games box scores of previous games. It yields an accuracy of 65.2% and employs Random Forest.
- "Sports Data Mining Technology Used In Basketball Outcome Prediction" is a project made by Cjenjie Cao from Dublin Institute of Technology. He includes 46 features that are the average of the teams' performance in the previous last 10 games. The accuracy percentage obtained in this project is 69.11% for the 2011 season.
- Bernard Loeffelholz tried to predict the NBA outcomes using Artificial Neural Networks. Although he yields an accuracy of 73.33%, the project might not be the best model to be compared with. The training data set only includes the 650 first games of the 2007 season, and the test data set is obtained from only 30 games. This amount is not enough to make sure that the percentage achieved will be maintained over time.
- "Accuscore" is a predictor developed by ESPN that yielded a 70.3% in the 2013 season.

Lastly, regarding the second experiment of this project, which tries to predict first team that will score, there are no projects related to the topic. The only assumption that can be done is provided by the bookmakers, who make us think that the probability for a team to score first, is the same as the one obtained in a heads or tails game.

Development

• Software used

To store the data, the software used was a simple Excel Sheet. Among the reasons for using this tool, we find the ease to apply filters to the dataset. Also, some features are calculated from others, and on Excel Sheets it is simple to use formulas that quickly generate these new features every time a new instance is added to the dataset.

The machine learning tool for this project is Weka. The program is written in Java, and it offers several features such as data pre-processing tools, classification tools, regression tools, clustering tools and features visualization.

Lastly, the tool MLP – SPN, also written in Java, was used to set the optimal neural network parameters that adjust the technique to the problem.

• Data extraction

The main problem in this part of the project is that the nature of the data set does not allow us to create a totally automatized phase. The origin of the different attributes are unalike, and the consults have been realized through different zones of the portal.

The process of getting data for the experiment "First Scorer" was especially complex. To get the information, it was necessary to consult every

play-by-play sheet of every match and check all the information, which will be explained in the Feature Selection section.

The reliable sources of information used were Basketball-Reference and OddsPortal.

• **Feature selection – Study of relevance.**

Two Weka algorithms were used to understand the correlation between all the features and the class:

- Chi Squared Attribute Evaluation: This test is used to show whether the occurrence of an attribute is independent of the occurrence of the class. Values close to 1 show that the correlation between the features and the class is high.
- Gain Ratio Attribute Evaluation: This test is applied to each feature, measuring its information gain ratio with respect to the class.

Once both evaluations are applied to both experiments, the features selected will be those which have a score higher than 0 on both evaluations.

• **Relevant features for "winner" experiment:**

The scores of all the attributes for the first experiment are listed below. The attributes represent aggregate values for the whole 2015/2016 season, which means that 1230 instances were collected. The letter L is applied to Local teams, and the letter V is applied to Visiting teams. Also, the OP indicator means that the attribute measures the performance of the opponents against that team.

Feature	Test Chi Square	Test Gain Ratio
Visitor Bet Line	244,11765	0,0774
Local Bet Line	240,22246	0,0724
Win ratio L	88,45034	0,0502
Win ratio V	87,38121	0,0381
Perc. Field Goals L OP	54,64725	0,0358
Points Per Game V	50,77163	0,0302
3P Attempts L OP	50,55388	0,0321
Recieved Points Per Game L	50,17949	0,0286
Points Per Game L	49,15564	0,0293
Perc. Field Goals L	49,10119	0,0726
Assists L OP	45,99982	0,029
Perc. Field Goals V OP	44,73097	0,0281
Field Goals L OP	41,51591	0,026
Field Goals V	39,05373	0,0484
Perc. 3P-shots L	38,62602	0,0247
Received Points Per Game V	37,25161	0,0233
Assists V OP	37,13952	0,0232
Assists L	37,04918	0,0578
Perc. Tiros V	36,19667	0,0489
3P-shots Anotados L	35,90695	0,0469
Offensive Rebounds L OP	35,72441	0,0143
3P-shots made L OP	33,72908	0,0321
Field Goals L	33,2273	0,0549
Perc. 3P-shots V	31,73148	0,0828
Rebounds Totales L	31,69675	0,0231
Rebounds Totales L OP	30,64306	0,0318
3P-shots Anotados V	30,3556	0,0888
Blocks L OP	29,79615	0,0417
Blocks V OP	29,37938	0,0191
3P-shots made V OP	28,96369	0,0779
Assists V	28,94212	0,0861
Total Rebounds V	24,78631	0,0155
Field Goals V OP	23,3422	0,0146
Perc. 3P-shots L OP	22,44517	0,0301
Turnovers L	22,08019	0,0329
3P-shots Attempts V	22,00452	0,0291
3P-shots Attempts V OP	20,25212	0,0126
3P-shots Attempts L	19,56174	0,0147
Total Rebounds V OP	19,15402	0,0151
Field Goals Attempts L OP	18,54014	0,0175
Perc. 3P-shots V OP	17,49824	0,0113
Free Throws made L OP	17,37856	0,109
Personal Fouls V OP	14,99608	0,011
Perc. Free Throws V OP	13,13831	0,1028
Streak V	12,54776	0,0271
Personal Fouls L	11,73478	0,1004
Blocks L	0	0
Steals L	0	0
Streak L	0	0
Field Goals Attempts L	0	0
Free Throws made L	0	0
Perc. Free Throws L	0	0
Offensive Rebounds L	0	0
Free Throw Attempts L	0	0
Free Days V	0	0
Free Throw Attempts L OP	0	0
Field Goals-Attempts V OP	0	0
Personal Fouls V	0	0
Blocks V	0	0
Free Throws Made V OP	0	0
Free Throws Attempts V OP	0	0
Offensive Rebounds V OP	0	0
Steals V OP	0	0
Turnovers V OP	0	0
Turnovers V	0	0
Steals V	0	0

Perc. Free Throws L OP	0	0
Free Days L	0	0
Steals L OP	0	0
Offensive Rebounds V	0	0
Personal Fouls L OP	0	0

Field Goals Attempts V	0	0
Free Throws Made V	0	0
Free Throws Attempts V	0	0
Perc. Free Throws V	0	0
Steals L OP	0	0

• Relevant features for experiment "First Scorer"

Due to the complexity involved in obtaining new examples for the techniques, the data set for this experiment consists in 302 instances, corresponding to the first 302 matches of the 2015/2016 season. The class of these instances is "First Scorer", which can be labeled as "Local" or "Visitor". The attributes collected are: Percentage of games where the team had the first possession (For local and for visitor), Percentage of accuracy on the first field goal (for local and for visitor) and percentage of times that the team has been the first scorer (for local and for visitor).

When the Chi Test and the Gain Ratio test are applied to these instances, we find that the score for all the attributes is 0. This result shows a low correlation between the features and the class, therefore, the accuracy percentage of this experiment will not be higher than 55%. In order to improve this experiment, the features of the "winner experiment" have been added to the instances of this current experiment. Despite the relevance test shows a score of 0 for all the attributes, the experiment will resume in order to confirm the low accuracy percentage.

• Data Pre-processing

Three pre-processing methods have been applied to the final data sets:

- Removing the irrelevant instances: The last 1161 games have been stored out of the 1230 original dataset. The reason for removing the

first 69 games is that the aggregated information that those games offers does not give us enough information to be processed by the techniques.

- Removing irrelevant attributes: All the attributes that scored 0 in both relevance tests have been removed.
- Normalization: All the values of all the features in the dataset have been normalized in the [0, 1] range. The formula used to achieve this effect is:

$$V' = \frac{\mathrm{V} - \min(\mathrm{V})}{\max(V) - \min(V)}$$

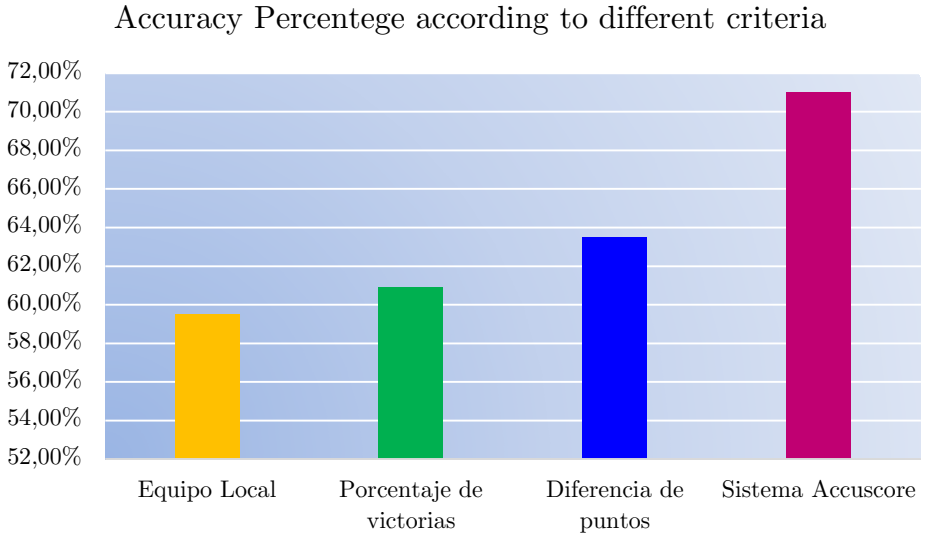
Where V is the original attribute and V' is the normalized attribute.

Experimentation and Results

- **Previous experimentation**

This icebreaker section will show the accuracy percentage for the experiment "Winner" using simple analysis.

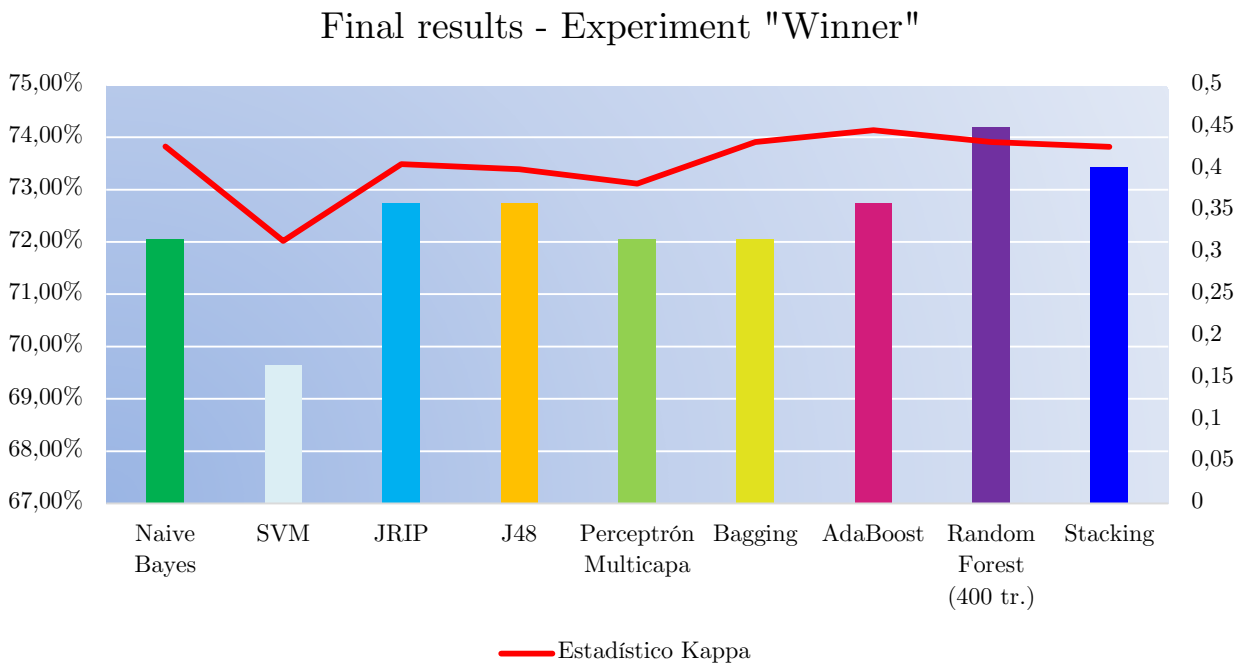
First of all, 59.51% of the games are won by the local team. 60.9% of the games are won by the team with the higher win ratio over the season. 63.5% of the games are won by the team with higher points per game over the season. Finally, it is significant that Accuscore is able to correctly predict 70.3% of the games.



• **"Winner experiment"**

For this experiment, the 1161 instances are split on a training set formed by 75% of the games, and a test set made by the last 25% matches of the season. The results obtained applying the different techniques are:

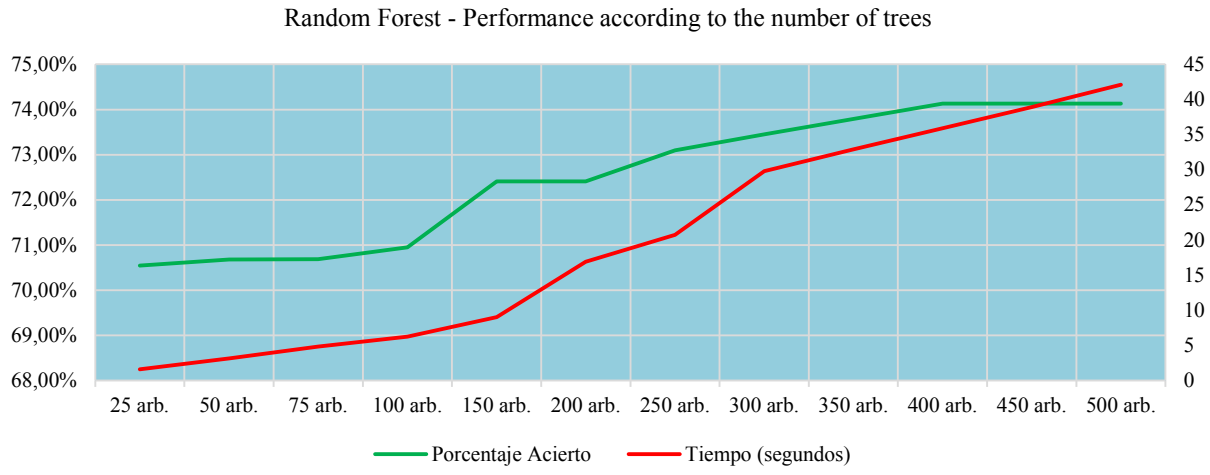
Experiment: Winner		
	Accuracy percentage	**Kappa Statistic**
Naïve Bayes	72,06%	0,4265
SVM	69,65%	0,3137
JRIP	72,75%	0,4054
J48	72,75%	0,3994
Multilayer Perceptron	72,06%	0,3821
Bagging	72,06%	0,4319
AdaBoost	72,75%	0,4459
Random Forest (400 tr.)	74,13%	0,4317
Stacking	73,44%	0,4262



.

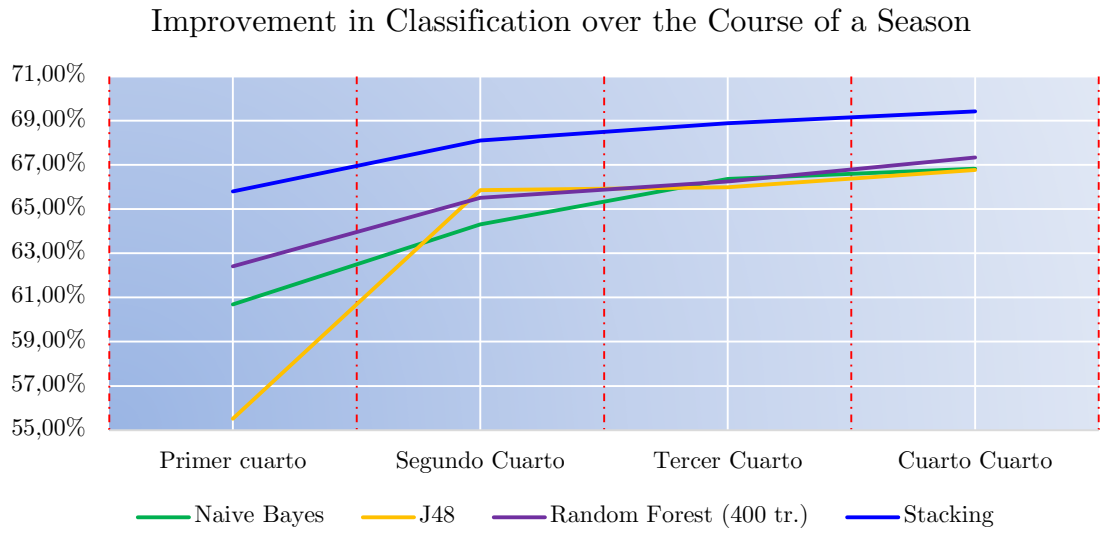
As can be seen, Random Forest with 400 trees and 7 random features is the technique with the highest accuracy percentage. This means that 215 matches out of 290 have been correctly classified.

The next chart shows the evolution of the Random Forest performance according to the number of trees implemented:



Another interesting piece of information obtained is the evolution of the accuracy percentage over the season for 4 of the best models generated. In this case, the test will be implemented under the "Cross-Validation" technique.

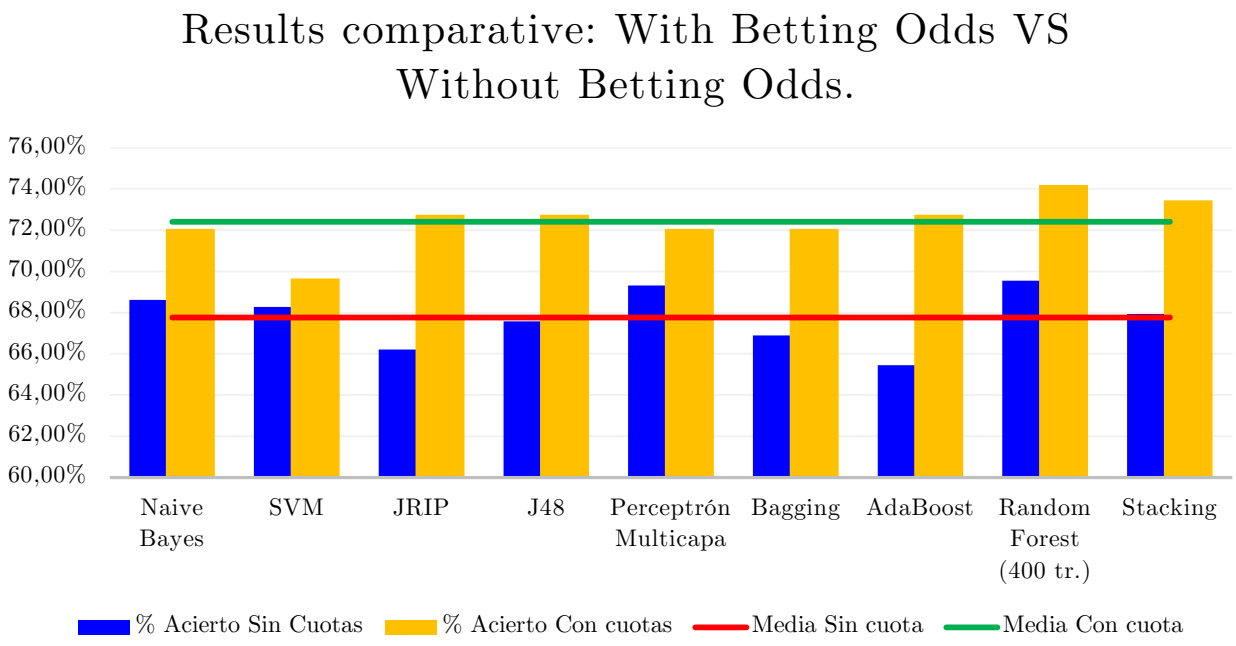
Mejora de la clasificación durante la temporada				
	Primer cuarto	Segundo Cuarto	Tercer Cuarto	Cuarto Cuarto
Naive Bayes	60,68%	64,31%	66,36%	66,83%
J48	55,51%	65,86%	65,99%	66,77%
Random Forest (400 tr.)	62,41%	65,51%	66,24%	67,33%
Stacking	65,80%	68,10%	68,88%	69,42%



It has been proved that the best results are obtained during the last quarter of the season. Moreover, it appears that the tendency of the accuracy percentage still grows over the time. Therefore, if the season were much longer, the team's performances would stabilize its mean, and the accuracy percentages would increase notably.

Another experiment that has been done is to exclude the attributes that are not essentially sport-related. In order to achieve this, the Betting Odds have been excluded, and the results are noticeably lower.

The best technique is, again, Random Forest with an accuracy percentage of 69.54%. This is almost 5% less than the best result including all the attributes.



• Comparative between "Winner" experiment results and other projects.

The 74.19% accuracy percentage yield in this project must be compared with other works, keeping in mind that the contexts for the other projects may differ from this project.

Accuscore achieved 70.3% accuracy in the 2013 season. The difference with this model is 3.89%. Despite the fact that a large part of the attributes used in both projects might be the same, the values are not. The differences between the 2013

season and the 2015 season could be very different due to different reasons such as changes in a team's performances.

The project "Predicting NBA Winners" yields an accuracy percentage of 65.2%. In this project only 17 attributes were used, and none of them were the Betting Odds. Therefore, it would be fairer to compare it with the 69.5% yielded in this project when Betting Odds are not included. The difference is still around 4% higher, but we cannot forget that the seasons are not the same.

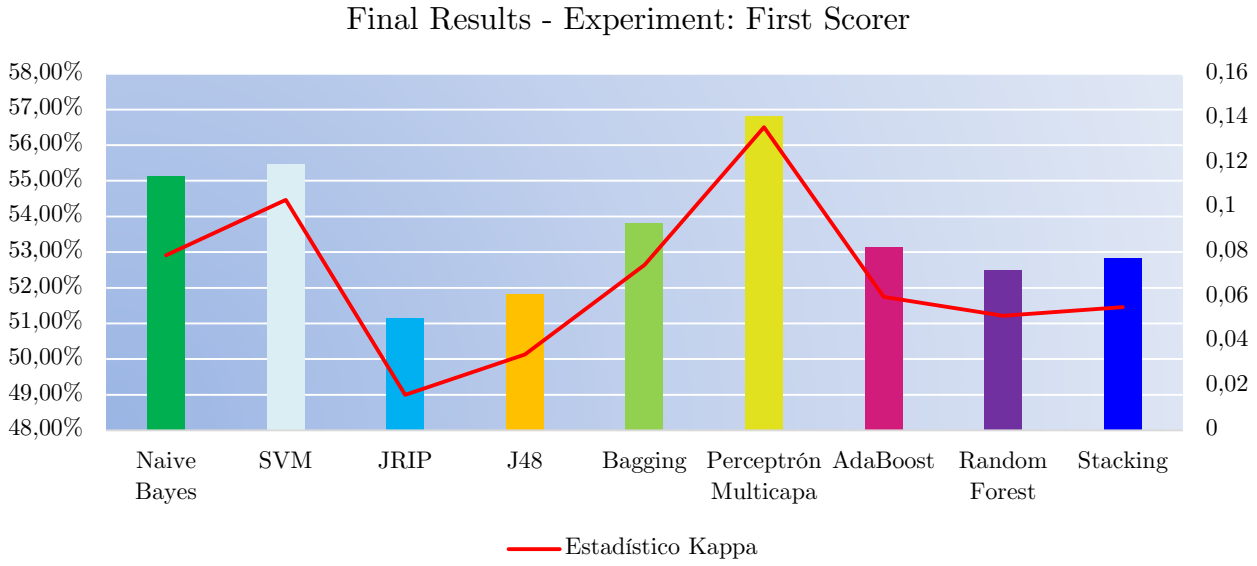
The Cao C. Project performed at 69.67%. This percentage is slightly higher than the 69.5% reached in this project when Betting Odds are not included. It is important to note that this project based their dataset on the performance of the last 10 NBA games before the prediction day. This tells us that the relevant information for predicting NBA outcomes is produced during the same season for which we are making predictions, especially if the information comes from the most recent games.

Lastly, the Loeffelhoz Project yielded 74.33%. This percentage is very close to the 74.19% obtained in this project. Nevertheless, it is important to remember that the test dataset used by Loeffelhoz is made up of only 30 games. If we look at appendix 4, the games played between April 3, 2016 and April 8, 2016 produced 83.33% of games correctly classified. In conclusion, larger test dataset must be used to make fair predictions in this specific area.

• **"First scorer" experimentation result.**

The best accuracy percentage obtained for this experiment is 56.8%, performed by the neural network. As was predicted during the studies of relevance, the model is not good enough to be utilized in a real context.

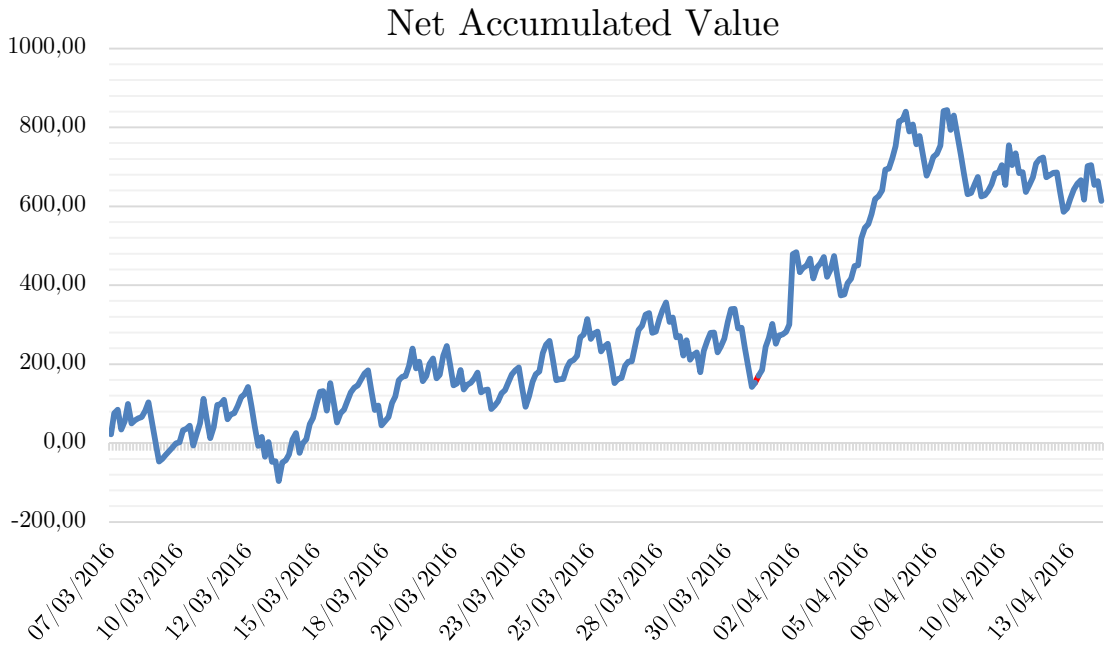
	Perc. Accuracy	**Kappa statistic**
Naive Bayes	55,14%	0,0786
SVM	55,48%	0,1035
JRIP	51,16%	0,0159
J48	51,82%	0,034
Neural network	56,81%	0,1361
Bagging	53.82%	0.0743
AdaBoost	53,15%	0,0599
Random Forest	52,49%	0,0514
Stacking	52,82%	0,0553



Nevertheless, it is important to note that only 302 instances have been applied, and it would be interesting to redo the experiment with the complete season.

Utilization of the best model

The best model (Random Forest) has been used to simulate its betting performance. All the information generated in this experiment has been collected in appendix 4. Every row of that appendix represents a game. It includes the game, the date, the net value obtained, the success probability and the net accumulated value until that game. The appendix shows that in the last game (ID 290), the net accumulated value is 122.7€. This benefit is obtained by betting 10€ on every prediction made by the model.



Nevertheless, the higher the amount of money bet, the higher the benefit. For example, if 50€ is bet on every prediction made by the model, the benefit rises to 613.5€.

To understand this, appendix 4 must be analyzed. The average Betting Odds of the games correctly classified is 1.40. On the other hand, the incorrectly classified/correctly classified ratio is 75/215=0,348, which means that if the Betting

Odds of the games correctly classified were 1,348, the net benefit would be 0. As the actual average is 1.40, which is higher than 1.348, the model generated in this project is good enough to generate profit for the last quarter of the 2015/2016 season.

Conclusions

Machine learning techniques have given us answers for two questions, and several interesting conclusions. The ones referring to the "winner" problem will be listed first, and the conclusions obtained for the "first scorer" problem will be explained afterwards.

After collecting, and pre-processing the 1230 games played during the season 2015/2016, the final dataset is formed by 1161 instances. The first 871 were used to train the models, and the last 290 were used to test them.

Among all the techniques used, Random Forest yielded the best accuracy percentage (74.13%). This means that 215 games out of 290 have been correctly classified, and 75 have not. The optimal model configuration was 7 random attributes (the approximated square root of the number of features) and 400 trees. For larger amounts of trees, the improvements on the results are insignificant.

This accuracy percentage is not high compared with other machine learning problems. For the test data set used, 3 out of 4 games are correctly classified, which means that it is still possible to improve the models. Nevertheless, the human behavior is a key factor in this problem, and it is not completely predictable.

On the other hand, it was proved that accuracy percentage grows over the season. The results obtained during first and second quarters of the season are poor

compared with the percentages achieved over the last games of the season. This concludes that if seasons were notably longer, teams performances would stabilized, and predictions would be easier for the models.

The accuracy percentages were considerably lower when only sport-related attributes were applied on the models. The best technique has been Random Forest, which has performed at 69.54%. This result gives a huge relevance to the Bet Lines attributes.

Comparisons between this project and others are not completely fair. Even though the accuracy percentage of this project is the highest, the others work over different seasons, and they do not include the Betting Lines as attributes. Therefore, one of the main conclusions of this project is: Whether only sport-related attributes are employed, the best results are yielded only if the included attributes measure the team's performances over the last games before the prediction date. Moreover, this predictions will hardly be higher than 69%. Therefore, if only sport-related attributes are wanted to be used, projects will have to investigate further. Attributes that are harder to obtain have to be collected, such as the frequent shot zones of every player, or the individual attack performances according to the players that are defending them.

This project includes the utilization of its best model over betting lines. The incorrect/correct ratio obtained with the best model is 75/215=0.348. This implies that if the odds mean of the 215 games were equal to 1.348, the net value would be 0€. However, the odds mean of the correct classified games is 1.40. This means that the generated model yields a positive net value for the last quarter of the 2015/2016 season.

Lastly, regarding the "first scorer" experiment, the results were poor. A Multilayer Perceptron, with one hidden layer and 20 neurons, has achieved 56.81% accuracy. This result was yielded applying cross-validation over 302 instances. This difference with 50% invites us to consider that a larger amount of attributes will increase the accuracy percentage to a certain extent. The final dataset has included all the attributes used for the "winner" problem. Therefore, even though the probabilities of being the first scorer are almost the same for both teams, the one that is favorite to win the game, is slightly favorite to be the first scorer.

Future works

As it has been explained, further improvements could still be made in the winner experiment. However, this experiment is surrounded by unpredictable facts and human behaviors. In this project, it has been assumed that the measurement of the mean performances of the teams is enough to develop the models. Nevertheless, further analysis over the individual performances can provide new approaches. The players tend to shoot from the same zone of the court. On the other hand, teams show better defensive skills on some particular zones of the court. Thus, the shot accuracy of the teams can be predicted more accurately by studying the zones and percentages of every individual player, and the defensive performances of the opposite team players on those zones. In other words, it is interesting to prove if the models would work better if individual attributes, which are not collected on the box scores, are supplied.

On the other hand, regarding the first scorer problem, the accuracy percentage can be improved to some extent. First of all, it is interesting to develop an automatic method that can collect the information extracted from the play-by-play

sheet. This step takes a big amount of time if it is done manually. Moreover, this improvement would allow to get a larger amount of instances, which could increase the accuracy percentage. In addition, the inclusion of individual performance attributes of the starting lineup is interesting.

Lastly, this experiments can be extrapolated to other basketball bet lines, such as winner before the half-time, amount of points scored by a particular player, or the total amount of points scored by both teams.

Anexo 2: Tabla de porcentajes para el problema de predicción: Primer Equipo en Anotar

Las siguientes tablas muestran los porcentajes recogidos manualmente partido a partido, para calcular el valor de los atributos del problema objetivo.

	ATLANTA	BOSTON	BROOKLYN	CHARLOTTE	CHICAGO
1er Pos SI	19	13	15	9	12
1er Pos NO	11	14	8	14	10
POS %	**0,633333333**	**0,481481481**	**0,652173913**	**0,391304348**	**0,545454545**
1er Shot IN	12	10	12	9	11
1er Shot OUT	14	17	11	14	11
SHOT %	**0,461538462**	**0,37037037**	**0,52173913**	**0,391304348**	**0,5**
1er en Anotar SI	10	15	15	11	13
1er en Anotar NO	16	12	8	12	9
First score %	**0,384615385**	**0,555555556**	**0,652173913**	**0,47826087**	**0,590909091**

	CLEVELAND	DALLAS	DENVER	DETROIT	GOLDEN STATE
1er Pos SI	8	11	7	18	14
1er Pos NO	14	14	18	8	11
POS %	**0,363636364**	**0,44**	**0,28**	**0,692307692**	**0,56**
1er Shot IN	13	11	13	11	12
1er Shot OUT	9	14	12	15	13
SHOT %	**0,590909091**	**0,44**	**0,52**	**0,423076923**	**0,48**
1er en Anotar SI	9	12	13	16	14
1er en Anotar NO	13	13	12	10	10
First score %	**0,409090909**	**0,48**	**0,52**	**0,615384615**	**0,583333333**

	HOUSTON	INDIANA	LAC	LAL	MEMPHIS
1er Pos SI	14	8	21	10	15
1er Pos NO	12	15	4	15	11
POS %	**0,538461538**	**0,347826087**	**0,84**	**0,4**	**0,576923077**
1er Shot IN	13	10	15	7	9
1er Shot OUT	13	13	10	18	16
SHOT %	**0,5**	**0,434782609**	**0,6**	**0,28**	**0,36**
1er en Anotar SI	10	11	14	9	12
1er en Anotar NO	16	12	11	16	13
First score %	**0,384615385**	**0,47826087**	**0,56**	**0,36**	**0,48**

	MIAMI	MILWAUKEE	MINNESOTA	NEW ORLEANS	NEW YORK
1er Pos SI	14	8	5	9	11
1er Pos NO	9	16	19	14	13
POS %	**0,608695652**	**0,333333333**	**0,208333333**	**0,391304348**	**0,458333333**
1er Shot IN	10	13	12	9	14
1er Shot OUT	14	11	12	14	10
SHOT %	**0,416666667**	**0,541666667**	**0,5**	**0,391304348**	**0,583333333**
1er en Anotar SI	13	10	18	11	15
1er en Anotar NO	11	14	6	12	9
First score %	**0,541666667**	**0,416666667**	**0,75**	**0,47826087**	**0,625**

	OKLAHOMA	ORLANDO	PHILADELPHIA	PHOENIX	PORTLAND
1er Pos SI	15	12	10	12	11
1er Pos NO	9	12	16	14	14
POS %	**0,625**	**0,5**	**0,384615385**	**0,461538462**	**0,44**
1er Shot IN	12	7	8	9	9
1er Shot OUT	12	17	18	17	16
SHOT %	**0,5**	**0,291666667**	**0,307692308**	**0,346153846**	**0,36**
1er en Anotar SI	16	11	7	11	12
1er en Anotar NO	8	13	19	15	13
First score %	**0,666666667**	**0,458333333**	**0,269230769**	**0,423076923**	**0,48**

	SACRAMENTO	SAN ANTONIO	TORONTO	UTAH	WASHINGTON
1er Pos SI	5	15	17	9	14
1er Pos NO	19	11	10	13	7
POS %	**0,208333333**	**0,576923077**	**0,62962963**	**0,409090909**	**0,666666667**
1er Shot IN	11	12	14	9	6
1er Shot OUT	13	14	13	13	15
SHOT %	**0,458333333**	**0,461538462**	**0,518518519**	**0,409090909**	**0,285714286**
1er en Anotar SI	7	13	18	11	13
1er en Anotar NO	16	13	9	11	8
First score %	**0,304347826**	**0,5**	**0,666666667**	**0,5**	**0,619047619**

Tabla 20: Cálculo de porcentajes para el experimento 2

Anexo 3: Árbol de decisión generado por J48 para el problema de clasificación de qué equipo ganara un partido determinado.

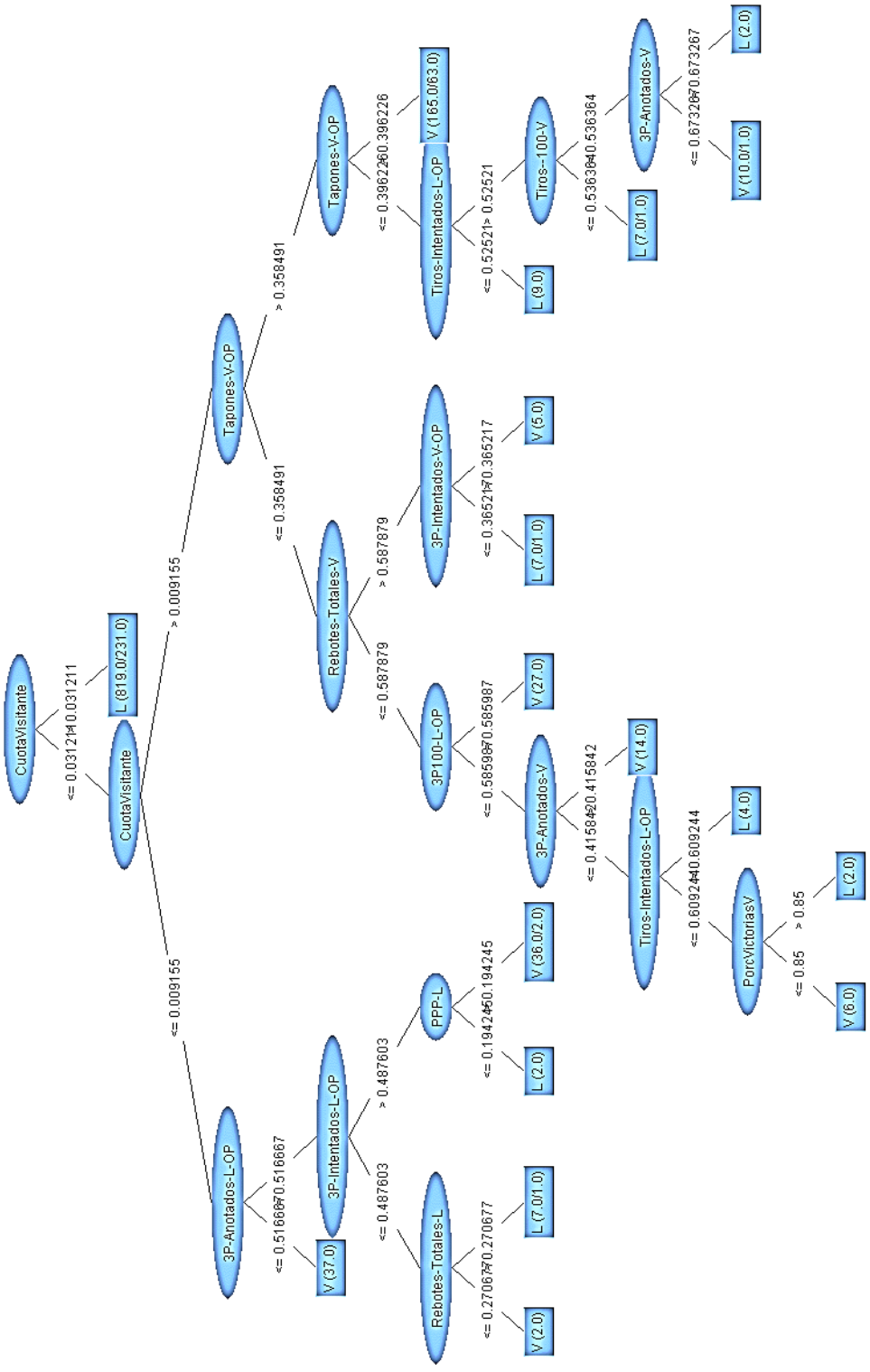


Ilustración 33: Árbol de decisión J48 - Experimento 1

Anexo 4: Tabla de predicción y beneficio del problema "ganador".

ID Partido	DATE	Partido	Cuota Local	Cuota Visitante	Predicción	Resultado Real	Benef. Neto por partido	Probabilidad acierto	BN Acumulado
1	07/03/2016	chi - mil	1,44	2,90	L	L	4,4	71,00%	4,40
2	07/03/2016	nop - sac	2,07	1,80	L	L	10,7	53,50%	15,10
3	07/03/2016	cho - min	1,18	5,14	L	L	1,8	80,80%	16,90
4	08/03/2016	lal - orl	2,41	1,59	V	L	-10	56,80%	6,90
5	08/03/2016	por - was	1,42	2,98	L	L	4,2	68,30%	11,10
6	08/03/2016	den - nyk	1,88	1,96	L	L	8,8	51,30%	19,90
7	08/03/2016	uta - atl	1,73	2,16	L	V	-10	67,80%	9,90
8	08/03/2016	min - sas	5,34	1,17	V	V	1,7	89,30%	11,60
9	08/03/2016	tor - brk	1,10	7,71	L	L	1	82,50%	12,60
10	09/03/2016	gsw - uta	1,05	11,19	L	L	0,5	95,00%	13,10
11	09/03/2016	sac - cle	3,38	1,34	V	V	3,4	76,80%	16,50
12	09/03/2016	okc - lac	1,41	3,02	L	L	4,1	52,80%	20,60
13	09/03/2016	pho - nyk	1,73	2,16	L	V	-10	52,80%	10,60
14	09/03/2016	dal - det	1,76	2,13	L	V	-10	61,80%	0,60
15	09/03/2016	mil - mia	2,40	1,60	V	L	-10	72,30%	-9,40
16	09/03/2016	bos - mem	1,13	6,37	L	L	1,3	79,50%	-8,10
17	09/03/2016	cho - nop	1,22	4,46	L	L	2,2	82,30%	-5,90
18	09/03/2016	phi - hou	5,04	1,19	V	V	1,9	88,80%	-4,00
19	10/03/2016	lal - cle	5,20	1,18	V	V	1,8	84,50%	-2,20
20	10/03/2016	den - pho	1,20	4,79	L	L	2	50,20%	-0,20
21	10/03/2016	sas - chi	1,05	12,11	L	L	0,5	97,30%	0,30
22	10/03/2016	tor - atl	1,62	2,38	L	L	6,2	64,50%	6,50
23	11/03/2016	gsw - por	1,07	9,85	L	L	0,7	95,80%	7,20
24	11/03/2016	lac - nyk	1,16	5,70	L	L	1,6	73,80%	8,80
25	11/03/2016	sac - orl	1,84	2,00	L	V	-10	65,80%	-1,20
26	11/03/2016	uta - was	1,55	2,54	L	L	5,5	61,30%	4,30
27	11/03/2016	chi - mia	2,48	1,57	V	V	5,7	68,00%	10,00
28	11/03/2016	mem - nop	2,24	1,68	L	L	12,4	54,50%	22,40
29	11/03/2016	okc - min	1,09	7,94	L	V	-10	85,00%	12,40
30	11/03/2016	bos - hou	1,50	2,67	L	V	-10	80,00%	2,40
31	11/03/2016	cho - det	1,57	2,47	L	L	5,7	63,30%	8,10
32	11/03/2016	phi - brk	2,13	1,75	L	L	11,3	54,00%	19,40
33	12/03/2016	gsw - pho	1,02	19,23	L	L	0,2	93,80%	19,60
34	12/03/2016	por - orl	1,24	4,27	L	L	2,4	80,80%	22,00
35	12/03/2016	den - was	1,88	1,97	V	L	-10	55,00%	12,00
36	12/03/2016	sas - okc	1,25	4,18	L	L	2,5	92,00%	14,50
37	12/03/2016	atl - mem	1,06	10,29	L	L	0,6	66,80%	15,10
38	12/03/2016	mil - nop	1,37	3,21	L	L	3,7	74,50%	18,80
39	12/03/2016	cho - hou	1,46	2,83	L	L	4,6	60,30%	23,40

40	12/03/2016	phi - det	6,53	1,13	V	V	1,3	82,00%	24,70
41	12/03/2016	tor - mia	1,38	3,18	L	L	3,8	79,50%	28,50
42	12/03/2016	dal - ind	1,88	1,96	L	V	-10	55,00%	18,50
43	13/03/2016	lal - nyk	2,07	1,79	L	V	-10	61,80%	8,50
44	13/03/2016	brk - mil	2,05	1,80	L	V	-10	63,00%	-1,50
45	13/03/2016	atl - ind	1,46	2,80	L	L	4,6	78,30%	3,10
46	13/03/2016	sac - uta	2,00	1,85	L	V	-10	55,00%	-6,90
47	13/03/2016	lac - cle	2,15	1,74	V	V	7,4	59,80%	0,50
48	14/03/2016	cho - dal	1,43	2,92	L	V	-10	71,50%	-9,50
49	14/03/2016	gsw - nop	1,03	14,14	L	L	0,3	97,00%	-9,20
50	14/03/2016	uta - cle	3,26	1,36	V	L	-10	59,30%	-19,20
51	14/03/2016	pho - min	1,93	1,91	L	L	9,3	52,50%	-9,90
52	14/03/2016	hou - mem	1,12	6,82	L	L	1,2	59,00%	-8,70
53	14/03/2016	okc - por	1,29	3,73	L	L	2,9	64,80%	-5,80
54	14/03/2016	was - det	1,77	2,10	L	L	7,7	64,50%	1,90
55	14/03/2016	mia - den	1,31	3,59	L	L	3,1	68,30%	5,00
56	14/03/2016	tor - chi	1,19	4,79	L	V	-10	81,00%	-5,00
57	15/03/2016	lal - sac	2,67	1,49	V	V	4,9	55,50%	-0,10
58	15/03/2016	sas - lac	1,20	4,84	L	L	2	69,50%	1,90
59	15/03/2016	mil - tor	2,08	1,77	V	V	7,7	57,30%	9,60
60	15/03/2016	brk - phi	1,31	3,60	L	L	3,1	67,80%	12,70
61	15/03/2016	ind - bos	1,70	2,21	L	L	7	56,30%	19,70
62	15/03/2016	orl - den	1,63	2,35	L	L	6,3	52,00%	26,00
63	16/03/2016	gsw - nyk	1,04	13,43	L	L	0,4	93,00%	26,40
64	16/03/2016	sac - nop	1,65	2,32	L	V	-10	61,80%	16,40
65	16/03/2016	hou - lac	1,60	2,40	V	V	14	51,50%	30,40
66	16/03/2016	mem - min	2,41	1,60	L	V	-10	57,00%	20,40
67	16/03/2016	det - atl	1,95	1,90	L	V	-10	51,80%	10,40
68	16/03/2016	bos - okc	2,77	1,47	V	V	4,7	58,50%	15,10
69	16/03/2016	cho - orl	1,18	5,28	L	L	1,8	78,30%	16,90
70	16/03/2016	cle - dal	1,42	2,97	L	L	4,2	71,50%	21,10
71	16/03/2016	was - chi	1,46	2,81	L	L	4,6	73,80%	25,70
72	17/03/2016	uta - pho	1,24	4,29	L	L	2,4	65,00%	28,10
73	17/03/2016	sas - por	1,12	6,99	L	L	1,2	97,50%	29,30
74	17/03/2016	atl - den	1,28	3,88	L	L	2,8	62,80%	32,10
75	17/03/2016	chi - brk	1,32	3,54	L	L	3,2	73,30%	35,30
76	17/03/2016	mil - mem	1,15	5,72	L	L	1,5	60,30%	36,80
77	17/03/2016	mia - cho	1,53	2,59	L	V	-10	65,00%	26,80
78	17/03/2016	ind - tor	1,79	2,06	L	V	-10	51,30%	16,80
79	17/03/2016	phi - was	4,54	1,22	V	V	2,2	86,80%	19,00
80	18/03/2016	lal - pho	1,59	2,44	L	V	-10	71,00%	9,00
81	18/03/2016	dal - gsw	4,72	1,21	V	V	2,1	68,50%	11,10
82	18/03/2016	hou - min	1,20	4,78	L	L	2	77,80%	13,10

83	18/03/2016	nop - por	2,18	1,72	V	V	7,2	50,70%	20,30
84	18/03/2016	det - sac	1,33	3,43	L	L	3,3	51,00%	23,60
85	18/03/2016	tor - bos	1,83	2,02	L	L	8,3	71,00%	31,90
86	18/03/2016	orl - cle	5,33	1,17	V	V	1,7	78,30%	33,60
87	18/03/2016	phi - okc	15,80	1,03	V	V	0,3	85,50%	33,90
88	19/03/2016	sas - gsw	1,53	2,60	L	L	5,3	75,00%	39,20
89	19/03/2016	chi - uta	1,87	1,96	L	L	8,7	63,30%	47,90
90	19/03/2016	mem - lac	5,11	1,18	V	L	-10	75,30%	37,90
91	19/03/2016	atl - hou	1,34	3,39	L	L	3,4	61,30%	41,30
92	19/03/2016	mia - cle	2,48	1,57	V	L	-10	65,80%	31,30
93	19/03/2016	det - brk	1,26	4,09	L	L	2,6	70,00%	33,90
94	19/03/2016	ind - okc	2,37	1,62	V	V	6,2	67,80%	40,10
95	19/03/2016	was - nyk	1,27	3,98	L	L	2,7	65,50%	42,80
96	19/03/2016	cho - den	1,20	4,88	L	V	-10	74,00%	32,80
97	20/03/2016	tor - orl	1,18	5,06	L	L	1,8	80,00%	34,60
98	20/03/2016	dal - por	1,97	1,87	L	L	9,7	61,30%	44,30
99	20/03/2016	nyk - sac	2,71	1,49	V	V	4,9	60,50%	49,20
100	20/03/2016	mil - uta	2,01	1,82	L	V	-10	51,30%	39,20
101	20/03/2016	nop - lac	5,36	1,17	V	L	-10	75,50%	29,20
102	20/03/2016	phi - bos	7,70	1,10	V	V	1	87,30%	30,20
103	21/03/2016	pho - mem	2,23	1,69	V	V	6,9	72,80%	37,10
104	21/03/2016	atl - was	1,33	3,45	L	V	-10	60,50%	27,10
105	21/03/2016	chi - sac	1,25	4,13	L	L	2,5	57,50%	29,60
106	21/03/2016	min - gsw	8,25	1,09	V	V	0,9	73,00%	30,50
107	21/03/2016	bos - orl	1,23	4,30	L	L	2,3	90,50%	32,80
108	21/03/2016	det - mil	1,29	3,70	L	L	2,9	71,50%	35,70
109	21/03/2016	cho - sas	3,04	1,40	V	L	-10	66,80%	25,70
110	21/03/2016	cle - den	1,11	7,35	L	L	1,1	78,50%	26,80
111	21/03/2016	ind - phi	1,04	12,70	L	L	0,4	93,80%	27,20
112	22/03/2016	lal - mem	2,33	1,63	V	L	-10	71,00%	17,20
113	22/03/2016	nop - mia	5,20	1,18	V	V	1,8	80,30%	19,00
114	22/03/2016	okc - hou	1,21	4,65	L	L	2,1	74,00%	21,10
115	22/03/2016	brk - cho	3,02	1,41	V	V	4,1	58,00%	25,20
116	23/03/2016	gsw - lac	1,16	5,60	L	L	1,6	86,50%	26,80
117	23/03/2016	pho - lal	1,43	2,91	L	L	4,3	59,00%	31,10
118	23/03/2016	por - dal	1,36	3,23	L	L	3,6	71,30%	34,70
119	23/03/2016	den - phi	1,22	4,45	L	L	2,2	74,30%	36,90
120	23/03/2016	sas - mia	1,14	5,94	L	L	1,4	94,50%	38,30
121	23/03/2016	chi - nyk	1,21	4,56	L	V	-10	72,80%	28,30
122	23/03/2016	hou - uta	1,79	2,07	L	V	-10	63,00%	18,30
123	23/03/2016	min - sac	1,55	2,52	L	L	5,5	50,50%	23,80
124	23/03/2016	bos - tor	1,72	2,19	L	L	7,2	72,00%	31,00
125	23/03/2016	det - orl	1,40	3,05	L	L	4	63,00%	35,00

126	23/03/2016	cle - mil	1,12	6,62	L	L	1,2	68,30%	36,20
127	23/03/2016	was - atl	1,89	1,95	V	V	9,5	50,50%	45,70
128	24/03/2016	lac - por	1,42	2,97	L	L	4,2	70,30%	49,90
129	24/03/2016	okc - uta	1,19	4,87	L	L	1,9	78,00%	51,80
130	24/03/2016	brk - cle	5,19	1,18	V	L	-10	77,80%	41,80
131	24/03/2016	nyk - chi	2,44	1,58	V	L	-10	56,50%	31,80
132	24/03/2016	ind - nop	1,05	11,88	L	L	0,5	85,30%	32,30
133	25/03/2016	gsw - dal	1,02	16,55	L	L	0,2	96,50%	32,50
134	25/03/2016	lal - den	2,49	1,56	V	V	5,6	53,80%	38,10
135	25/03/2016	sac - pho	1,32	3,53	L	L	3,2	50,00%	41,30
136	25/03/2016	sas - mem	1,07	9,74	L	L	0,7	94,80%	42,00
137	25/03/2016	atl - mil	1,22	4,47	L	L	2,2	51,80%	44,20
138	25/03/2016	hou - tor	1,94	1,89	L	L	9,4	52,30%	53,60
139	25/03/2016	mia - orl	1,15	5,83	L	L	1,5	75,00%	55,10
140	25/03/2016	det - cho	1,77	2,09	L	L	7,7	63,00%	62,80
141	25/03/2016	was - min	1,23	4,46	L	V	-10	62,00%	52,80
142	26/03/2016	min - uta	3,89	1,28	V	V	2,8	71,80%	55,60
143	26/03/2016	okc - sas	1,09	8,17	L	L	0,9	69,30%	56,50
144	26/03/2016	det - atl	1,86	1,97	L	V	-10	53,00%	46,50
145	26/03/2016	nyk - cle	4,18	1,24	V	V	2,4	80,00%	48,90
146	26/03/2016	nop - tor	5,93	1,14	V	V	1,4	70,30%	50,30
147	26/03/2016	orl - chi	3,37	1,34	V	L	-10	54,00%	40,30
148	26/03/2016	brk - ind	2,99	1,42	V	L	-10	66,30%	30,30
149	26/03/2016	pho - bos	4,43	1,22	V	V	2,2	80,50%	32,50
150	26/03/2016	por - phi	1,04	13,29	L	L	0,4	88,00%	32,90
151	26/03/2016	mil - cho	2,35	1,63	V	V	6,3	57,50%	39,20
152	27/03/2016	lal - was	4,61	1,21	V	V	2,1	81,30%	41,30
153	27/03/2016	gsw - phi	1,01	23,43	L	L	0,1	93,80%	41,40
154	27/03/2016	ind - hou	1,83	2,00	L	L	8,3	51,50%	49,70
155	27/03/2016	sac - dal	1,76	2,11	L	L	7,6	62,80%	57,30
156	27/03/2016	lac - den	1,22	4,35	L	L	2,2	69,50%	59,50
157	28/03/2016	lac - bos	1,56	2,51	L	L	5,6	68,80%	65,10
158	28/03/2016	por - sac	1,08	8,93	L	L	0,8	74,80%	65,90
159	28/03/2016	den - dal	1,78	2,08	L	V	-10	59,30%	55,90
160	28/03/2016	uta - lal	1,06	10,55	L	L	0,6	86,50%	56,50
161	28/03/2016	chi - atl	2,33	1,63	V	V	6,3	62,30%	62,80
162	28/03/2016	mem - sas	2,80	1,46	V	V	4,6	74,00%	67,40
163	28/03/2016	min - pho	1,39	3,08	L	L	3,9	50,70%	71,30
164	28/03/2016	nop - nyk	2,90	1,43	V	L	-10	54,50%	61,30
165	28/03/2016	mia - brk	1,23	4,46	L	L	2,3	72,30%	63,60
166	28/03/2016	tor - okc	2,28	1,66	L	V	-10	58,80%	53,60
167	29/03/2016	gsw - was	1,07	9,25	L	L	0,7	94,50%	54,30
168	29/03/2016	cle - hou	1,80	2,06	L	V	-10	54,50%	44,30

169	29/03/2016	det - okc	1,79	2,07	L	L	7,9	64,30%	52,20
170	29/03/2016	ind - chi	1,38	3,13	L	V	-10	78,50%	42,20
171	29/03/2016	orl - brk	1,26	4,07	L	L	2,6	67,30%	44,80
172	29/03/2016	phi - cho	7,06	1,11	V	V	1,1	86,80%	45,90
173	30/03/2016	lal - mia	5,32	1,12	V	L	-10	86,30%	35,90
174	30/03/2016	sac - was	2,08	1,77	L	L	10,8	53,30%	46,70
175	30/03/2016	uta - gsw	2,59	1,53	V	V	5,3	66,50%	52,00
176	30/03/2016	dal - nyk	1,39	3,10	L	L	3,9	73,00%	55,90
177	30/03/2016	sas - nop	1,01	25,05	L	L	0,1	96,50%	56,00
178	30/03/2016	mem - den	1,68	2,24	L	V	-10	66,80%	46,00
179	30/03/2016	mil - pho	1,29	3,73	L	L	2,9	65,50%	48,90
180	30/03/2016	min - lac	3,00	1,41	V	V	4,1	69,50%	53,00
181	30/03/2016	tor - atl	1,86	1,98	L	L	8,6	65,50%	61,60
182	31/03/2016	por - bos	1,62	2,36	L	L	6,2	60,50%	67,80
183	31/03/2016	okc - lac	1,03	15,68	L	L	0,3	74,00%	68,10
184	31/03/2016	nop - den	3,45	1,33	V	L	-10	59,30%	58,10
185	31/03/2016	cle - brk	1,03	16,32	L	L	0,3	84,50%	58,40
186	31/03/2016	hou - chi	1,35	3,33	L	V	-10	77,00%	48,40
187	31/03/2016	ind - orl	1,38	3,19	L	V	-10	59,30%	38,40
188	01/04/2016	gsw - bos	1,10	6,10	L	V	-10	91,00%	28,40
189	01/04/2016	pho - was	4,51	1,22	V	V	2,2	81,50%	30,60
190	01/04/2016	sac - mia	3,34	1,35	V	V	3,5	80,50%	34,10
191	01/04/2016	uta - min	1,30	3,65	L	L	3	72,50%	37,10
192	01/04/2016	atl - cle	1,72	2,16	V	V	11,6	52,00%	48,70
193	01/04/2016	mem - tor	2,85	1,45	V	V	4,5	60,80%	53,20
194	01/04/2016	mil - orl	1,72	2,18	L	L	7,2	61,00%	60,40
195	01/04/2016	det - dal	1,44	2,91	L	V	-10	62,50%	50,40
196	01/04/2016	nyk - brk	1,41	3,03	L	L	4,1	65,50%	54,50
197	01/04/2016	cho - phi	1,05	11,50	L	L	0,5	92,00%	55,00
198	02/04/2016	phi - ind	6,44	1,13	V	V	1,3	89,00%	56,30
199	02/04/2016	por - mia	1,38	3,14	L	L	3,8	65,50%	60,10
200	02/04/2016	den - sac	1,23	4,57	V	V	35,7	52,50%	95,80
201	02/04/2016	sas - tor	1,09	8,38	L	L	0,9	96,50%	96,70
202	02/04/2016	chi - det	1,66	2,29	L	V	-10	63,00%	86,70
203	03/04/2016	lal - bos	4,71	1,21	V	V	2,1	79,00%	88,80
204	03/04/2016	gsw - por	1,12	6,90	L	L	1,2	91,30%	90,00
205	03/04/2016	nyk - ind	3,36	1,35	V	V	3,5	76,00%	93,50
206	03/04/2016	mil - chi	1,75	2,14	L	V	-10	74,00%	83,50
207	03/04/2016	orl - mem	1,55	2,55	L	L	5,5	51,80%	89,00
208	03/04/2016	pho - uta	4,88	1,20	V	V	2	83,30%	91,00
209	03/04/2016	cle - cho	1,33	3,45	L	L	3,3	74,30%	94,30
210	03/04/2016	hou - okc	2,31	1,65	V	L	-10	61,00%	84,30
211	03/04/2016	lac - was	1,37	3,26	L	L	3,7	63,30%	88,00

212	03/04/2016	min - dal	2,25	1,68	V	V	6,8	52,00%	94,80
213	03/04/2016	brk - nop	1,69	2,25	L	V	-10	64,80%	84,80
214	05/04/2016	gsw - min	1,04	13,87	L	V	-10	93,30%	74,80
215	05/04/2016	lac - lal	1,05	11,27	L	L	0,5	92,00%	75,30
216	05/04/2016	sac - por	2,49	1,57	V	V	5,7	60,80%	81,00
217	05/04/2016	den - okc	4,47	1,23	V	V	2,3	72,80%	83,30
218	05/04/2016	uta - sas	2,33	1,64	V	V	6,4	72,30%	89,70
219	05/04/2016	atl - pho	1,04	13,83	L	L	0,4	81,80%	90,10
220	05/04/2016	mem - chi	2,36	1,62	L	L	13,6	56,50%	103,70
221	05/04/2016	mia - det	1,54	2,55	L	L	5,4	56,80%	109,10
222	05/04/2016	mil - cle	4,92	1,20	V	V	2	78,00%	111,10
223	05/04/2016	tor - cho	1,49	2,72	L	L	4,9	70,80%	116,00
224	05/04/2016	phi - nop	1,76	2,13	L	L	7,6	66,50%	123,60
225	06/04/2016	lal - lac	5,94	1,15	V	V	1,5	83,30%	125,10
226	06/04/2016	por - okc	1,30	3,75	L	L	3	63,50%	128,10
227	06/04/2016	dal - hou	2,05	1,81	L	L	10,5	62,80%	138,60
228	06/04/2016	bos - nop	1,05	12,22	L	L	0,5	90,50%	139,10
229	06/04/2016	nyk - cho	2,60	1,53	V	V	5,3	66,30%	144,40
230	06/04/2016	ind - cle	1,61	2,41	L	L	6,1	57,30%	150,50
231	06/04/2016	orl - det	1,68	2,26	V	V	12,6	53,30%	163,10
232	06/04/2016	was - brk	1,09	8,17	L	L	0,9	68,80%	164,00
233	07/04/2016	gsw - sas	1,39	3,14	L	L	3,9	78,50%	167,90
234	07/04/2016	sac - min	1,95	1,89	L	V	-10	66,00%	157,90
235	07/04/2016	atl - tor	1,36	3,23	L	L	3,6	57,50%	161,50
236	07/04/2016	hou - pho	1,09	8,22	L	V	-10	66,50%	151,50
237	07/04/2016	mia - chi	1,41	3,02	L	L	4,1	72,30%	155,60
238	08/04/2016	den - sas	1,81	2,05	V	L	-10	63,30%	145,60
239	08/04/2016	uta - lac	1,07	9,62	L	V	-10	73,00%	135,60
240	08/04/2016	dal - mem	1,38	3,18	L	L	3,8	70,30%	139,40
241	08/04/2016	nop - lal	1,57	2,48	L	L	5,7	59,50%	145,10
242	08/04/2016	bos - mil	1,14	6,18	L	L	1,4	86,50%	146,50
243	08/04/2016	det - was	1,42	2,95	L	L	4,2	59,30%	150,70
244	08/04/2016	tor - ind	2,76	1,47	L	L	17,6	53,50%	168,30
245	08/04/2016	cho - brk	1,05	11,74	L	L	0,5	84,30%	168,80
246	08/04/2016	orl - mia	2,01	1,84	V	L	-10	58,00%	158,80
247	08/04/2016	phi - nyk	2,16	1,73	V	V	7,3	53,00%	166,10
248	09/04/2016	nop - pho	2,30	1,66	L	V	-10	58,30%	156,10
249	09/04/2016	por - min	1,24	4,30	L	V	-10	73,50%	146,10
250	09/04/2016	sac - okc	4,23	1,25	V	L	-10	87,50%	136,10
251	09/04/2016	chi - cle	3,51	1,33	V	L	-10	67,00%	126,10
252	09/04/2016	mem - gsw	9,39	1,07	V	V	0,7	65,50%	126,80
253	09/04/2016	atl - bos	1,43	2,91	L	L	4,3	66,00%	131,10
254	10/04/2016	nyk - tor	3,19	1,38	V	V	3,8	66,00%	134,90

255	10/04/2016	sas - gsw	1,63	2,35	L	V	-10	71,00%	124,90
256	10/04/2016	ind - brk	1,08	8,79	L	L	0,8	82,80%	125,70
257	10/04/2016	mia - orl	1,20	4,90	L	L	2	71,00%	127,70
258	10/04/2016	den - uta	3,34	1,35	V	V	3,5	69,80%	131,20
259	10/04/2016	phi - mil	2,58	1,54	V	V	5,4	50,70%	136,60
260	10/04/2016	hou - lal	1,06	10,53	L	L	0,6	79,00%	137,20
261	10/04/2016	lac - dal	1,37	3,18	L	L	3,7	70,30%	140,90
262	10/04/2016	was - cho	2,82	1,46	V	L	-10	50,70%	130,90
263	11/04/2016	pho - sac	1,41	3,00	V	V	20	51,00%	150,90
264	11/04/2016	uta - dal	1,34	3,43	L	V	-10	73,30%	140,90
265	11/04/2016	min - hou	2,41	1,60	V	V	6	51,30%	146,90
266	11/04/2016	nop - chi	2,74	1,49	L	V	-10	59,50%	136,90
267	11/04/2016	okc - lal	1,04	13,92	L	L	0,4	92,50%	137,30
268	11/04/2016	bos - cho	1,33	3,46	L	V	-10	82,30%	127,30
269	11/04/2016	brk - was	3,35	1,35	V	V	3,5	64,30%	130,80
270	11/04/2016	cle - atl	1,39	3,14	L	L	3,9	63,00%	134,70
271	11/04/2016	orl - mil	1,70	2,21	L	L	7	66,80%	141,70
272	12/04/2016	lac - mem	1,22	4,48	L	L	2,2	68,80%	143,90
273	12/04/2016	sas - okc	1,08	8,66	L	L	0,8	91,50%	144,70
274	12/04/2016	det - mia	1,94	1,90	L	V	-10	52,50%	134,70
275	12/04/2016	tor - phi	1,11	7,33	L	L	1,1	88,80%	135,80
276	12/04/2016	ind - nyk	1,12	6,76	L	L	1,2	81,50%	137,00
277	13/04/2016	gsw - mem	1,02	16,52	L	L	0,2	93,80%	137,20
278	13/04/2016	lal - uta	2,36	1,63	V	L	-10	67,80%	127,20
279	13/04/2016	pho - lac	1,50	2,67	V	L	-10	56,30%	117,20
280	13/04/2016	por - den	1,17	5,49	L	L	1,7	67,80%	118,90
281	13/04/2016	bos - mia	1,49	2,72	L	L	4,9	73,80%	123,80
282	13/04/2016	brk - tor	2,78	1,48	V	V	4,8	55,00%	128,60
283	13/04/2016	cho - orl	1,29	3,80	L	L	2,9	75,30%	131,50
284	13/04/2016	chi - phi	1,18	5,09	L	L	1,8	80,80%	133,30
285	13/04/2016	cle - det	1,56	2,51	L	V	-10	57,80%	123,30
286	13/04/2016	dal - sas	1,49	2,71	V	V	17,1	50,20%	140,40
287	13/04/2016	hou - sac	1,05	10,80	L	L	0,5	68,30%	140,90
288	13/04/2016	mil - ind	1,51	2,67	L	V	-10	53,30%	130,90
289	13/04/2016	min - nop	1,18	5,13	L	L	1,8	67,00%	132,70
290	13/04/2016	was - atl	4,56	1,22	V	L	-10	72,00%	122,70

Ilustración 34: Predicción del modelo para el experimento 1 - Calculo del Beneficio Neto